



Optimisation énergétique des protocoles de communication des réseaux de capteurs sans fil

Rivo Sitraka Andriambolamalala Randriatsiferana

► To cite this version:

Rivo Sitraka Andriambolamalala Randriatsiferana. Optimisation énergétique des protocoles de communication des réseaux de capteurs sans fil. Réseaux et télécommunications [cs.NI]. Université de la Réunion, 2014. Français. NNT : 2014LARE0019 . tel-01155659

HAL Id: tel-01155659

<https://theses.hal.science/tel-01155659>

Submitted on 27 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LA REUNION

Faculté des Sciences et Technologies

ECOLE DOCTORALE SCIENCES - TECHNOLOGIES - SANTE

DISCIPLINE : **Réseaux et Télécommunications**

Présentée par

Rivo Sitraka A. RANDRIATSIFERANA

**Optimisation énergétique des protocoles de
communication des réseaux de capteurs sans fil**

Composotion du jury :

Rapporteur	M. Abdelhamid MELLOUK, Professeur, Université Paris-Est Créteil (UPEC).
Rapporteur	M. Tahiry RAZAFINDRALAMBO, Chargé de recherche H.D.R, INRIA Lille-Nord Europe.
Président du jury	M. Abdelhamid MELLOUK, Professeur, Université Paris-Est Créteil (UPEC).
Dir. de thèse	M. Jean-Daniel LAN SUN LUK, Professeur, Université de La Réunion.
Examineur	M. Frédéric ALICALAPA, Maître de Conférence, Université de La Réunion.
Invité	M. Richard LORION, Maître de Conférence, Université de La Réunion.
Invité	M. Frédéric ORY, Responsable de Projet, Entreprise TEEO La Réunion.

Soutenue le 03/12/2014

Remerciements

Grâce à Dieu, j'ai l'honneur d'inscrire ici un immense remerciement à mes parents et toute ma famille, de m'avoir soutenu et encouragé pendant toutes mes études et surtout pendant ces trois années et huit mois loin de notre Pays.

J'exprime mes sincères gratitude envers l'ensemble des membres du jury. Tout d'abord, je remercie vivement Mr Abdelhamid MELLOUK, professeur, de l'Universitaire Paris Est (UPEC), et Mr Tahiry RAZAFINDRALAMBO, Chargé de recherche H.D.R, INRIA Lille-Nord Europe, de bien avoir accepté d'être rapporteurs de mes travaux. J'aimerais également témoigner ma gratitude à Monsieur Pr Abdelhamid MELLOUK de bien avoir endossé la charge de président du jury. Je remercie également Monsieur Frédéric ORY, responsable de projet TEEO La Réunion, pour leur participation à ce jury.

Mes remerciements les plus sincères s'adressent en premier lieu à notre Directeur du Laboratoire Pr. Jean-Pierre CHABRIAT et mon directeur de thèse Pr. Jean Daniel LAN SUN LUK, qui m'ont accueilli et donné l'opportunité de réaliser ce travail de thèse au sein du laboratoire LE²P. Je voudrais remercier également Mr Richard Lorion, maître de conférence, de l'université de la Réunion, et Mr Frederic Alicalapa, maître de conférence, de l'université de la Réunion, d'avoir encadré cette thèse, et en particulier Mr Frederic Alicalapa, en qui j'ai trouvé plus qu'un encadrant, un grand frère. A lui, mon admiration et mes remerciements les plus profonds. Ces années de travail en commun m'ont beaucoup appris scientifiquement et humainement. Je souhaite à tous les doctorants d'avoir des encadreurs de thèse aussi généreux à partager leur savoir, leur travail et leur temps.

Sans doute que de nombreux facteurs ont une incidence sur le succès et les résultats d'une thèse. La thèse n'est pas faite seulement entre un doctorant et ses encadrants. Les multiples discussions entre doctorants qui se génèrent et les relations amicales qui naissent avec les personnes qui travaillent dans le laboratoire et ailleurs, jouent un rôle tout aussi fondamental. Je remercie tout les membres du laboratoire LE²P, j'adresse particulièrement à Mr Alexandre Douère, maître de conférence, et Dr Pierre Olivier, bien sûr à mes collègues les futurs docteurs : Roche Sébastien et David Latchoumaia ainsi que tous les autres doctorants, qui m'ont toujours aidé et soutenu dans divers problèmes. Je tiens également à remercier Mr Fanilo HARIVELO, Maître de Conférences, Laboratoire LIM, de l'université de La Réunion, pour son soutien indéfectible dans les moments difficiles et sa sympathie durant le temps où on a travaillé ensemble et pour les discussions de recherche dans le domaine des réseaux sans fil.

J'adresse, particulièrement, mes remerciements les plus distingués à Mr. Swades De, pour sa gentillesse, son aide désintéressée et son énorme patience avec les multiples échanges que nous avons effectué au cours de mon séjour à IIT Delhi, India.

Table des matières

1	Introduction Générale	1
1.1	Motivation	1
1.2	Contributions	3
1.3	Structure de la thèse	4
2	Etat de l'art des réseaux de capteurs sans fil (RCSF)	7
2.1	Introduction	7
2.2	Description et caractéristiques du protocole de routage hiérarchique .	11
2.2.1	La description du réseau hiérarchique	11
2.2.2	Les caractéristiques du réseau hiérarchique	12
2.3	Les algorithmes de clustering	13
2.3.1	Les algorithmes de construction du cluster	13
2.3.2	Les modes de transmission des données	15
2.4	Présentation des principaux protocoles de routage hiérarchique . . .	17
2.4.1	Protocoles basés sur la construction du cluster	17
2.4.2	Protocoles de routage basés sur la transmission de données .	30
2.5	Conclusion	32
3	Optimisation des protocoles de communication	35
3.1	Introduction	36
3.2	Hypothèses sur le RCSF et définition des métriques	38
3.3	Formulation des problèmes	39
3.3.1	Coefficient de variation	40
3.3.2	Fonction Convexe	41
3.3.3	Description de modèle énergétique du noeud	42
3.3.4	Estimation de consommation d'énergie globale du réseau . . .	43
3.4	Les protocoles centralisés	46
3.4.1	Introduction	46
3.4.2	Limitations de LEACH et LEACH-C protocoles	46
3.4.3	Détermination de nombre de cluster optimal	47
3.4.4	Critère et choix du "cluster-head"	49
3.4.5	Algorithme du k-LEACH et son variant	50
3.4.6	Simulation et Analyse	52
3.4.7	Conclusion	57
3.5	Le protocole distribué : e-LEACH	58
3.5.1	Introduction et Motivation	58

3.5.2	Paramètres de sélection du "cluster-head"	58
3.5.3	Equilibrage de la consommation d'énergie	59
3.5.4	Simulation et Discussion	61
3.5.5	Conclusion	63
3.6	Le protocole décentralisé : LEACH-C2D	65
3.6.1	Introduction et Motivation	65
3.6.2	Sélection du "cluster-head" : Equilibrage d'énergie consommée	65
3.6.3	Limitation de l'équilibrage de la consommation d'énergie	68
3.6.4	Modèle énergétique global exponentiel	70
3.6.5	Résultats : Modèle de consommation globale exponentielle	71
3.6.6	Conclusion	74
3.7	Cluster-Based Multihop Protocols	75
3.7.1	Motivation	75
3.7.2	Forwarding and Routing Stateless Multi-Hop Protocol	76
3.7.3	Distributed Cluster-Based Routing Protocol	78
3.7.4	Simulations et Discussions	85
3.7.5	Conclusion	87
3.8	Conclusion de chapitre	87
4	Caractérisation des consommations d'énergie d'un noeud	89
4.1	Introduction	89
4.2	Plateforme utilisée pour l'implémentation	90
4.2.1	Le système d'exploitation	90
4.2.2	Les noeuds du réseau	91
4.3	Méthodologie expérimentale de mesure des consommations	92
4.3.1	Les blocs des composants fonctionnels	93
4.3.2	Systèmes de mesure des consommations	96
4.4	Les résultats de mesures des consommations	102
4.4.1	Le microcontrôleur	102
4.4.2	Le module Timer	104
4.4.3	Les Leds (diodes électroluminescentes)	105
4.4.4	Le module radio	106
4.4.5	La mémoire flash externe	111
4.5	Modélisation et synthèse	111
4.5.1	Le microcontrôleur	112
4.5.2	Le transceiver	113
4.5.3	Synthèse des énergies consommées	117
4.6	Conclusions	121

5 Conclusion générale et perspectives	123
5.1 Conclusions générales	123
5.1.1 L'algorithme de clustérisation centralisé	124
5.1.2 L'algorithme de clustérisation distribué	125
5.1.3 L'algorithme décentralisé	125
5.1.4 L'algorithme multihop des protocoles de routage	126
5.1.5 Caractérisation des consommations énergétiques d'un noeud .	126
5.2 Perspectives	127
5.2.1 Aspect protocolaire	127
5.2.2 Aspect expérimental	128
Bibliographie	129

Table des figures

2.1	communication intra-cluster et inter-cluster dans le réseau	12
2.2	Topologie : LEACH Protocol	18
2.3	Topologie : DWEHC Protocol	22
2.4	Topologie :Two-Level Hierarchy LEACH	23
2.5	Topologie : Unequal Clustering Size	24
2.6	Topologie :BCDCP protocol	29
2.7	Topologie :PEGASIS	31
3.1	Modèle de consommation d'énergie du module radio	42
3.2	Nombre optimal du cluster en fonction du paramètre alpha	53
3.3	Illustration de la durée de vie de k-LEACH et k-LEACH-VAR en fonction de rayon de couverture	54
3.4	Durée de vie du réseau : k-LEACH et K-LEACH-VAR	55
3.5	Energie restante globale du réseau : k-LEACH et k-LEACH-VAR	56
3.6	Quantités de données reçues par la station de base	56
3.7	Comparaison de la durée de vie d'e-LEACH	61
3.8	Comparaison d'énergie restante d'e-LEACH	62
3.9	Comparaison des traffics au sein du réseau	63
3.10	Modèle de consommation	68
3.11	Durée de vie : modèle d'énergie globale fonction linéaire décroissante	69
3.12	Energie restante à 300 tours.	70
3.13	Illustration de la durée de vie de k-LEACH et K-LEACH-C2D en fonction de rayon de couverture	72
3.14	Comparaison de la durée de vie de k-LEACH-C2D	73
3.15	Comparaison d'énergie restante de k-LEACH-C2D	73
3.16	Comparaison des traffics au sein du réseau k-LEACH-C2D	74
3.17	Architecture du réseau dans FRSM	77
3.18	Opération dans le protocole CB-RSM	80
3.19	Architecture du réseau : (A) spanning tree, (B)FRSM et (C) CB-RSM	81
3.20	Comparaison de la durée de vie : FRSM et CB-BGF	85
3.21	Comparaison d'énergie restante : FRSM et CB-BGF	86
3.22	Comparaison des traffics au sein du réseau :FRSM et CB-BGF	87
4.1	Noeud de réseau telosB [1]	92
4.2	Schéma synoptique de TI MSP430, Telob [1]	93
4.3	Mesure d'énergie avec la résistance shunt	98
4.4	schéma synoptique de mesure	99

4.5	Brochage de l'AD627	100
4.6	Montage pour relever de l'offset de tension	101
4.7	Circuit pour le calibrage par minimisation de courant d'offset - ajout de résistance d'entrée.	101
4.8	Schéma de mesure final à base de l'AD627 pour les mesures de consom- mation.	102
4.9	Msp430 :Courants consommés dans les différents LPMs.	104
4.10	Courant consommé : (a)Timer Périodique, (b)Timer One-shot	105
4.11	Consommation des LEDs	106
4.12	Phase d'émission mesurée, L=50 bytes, 0 dBm	107
4.13	Transfert mesuré entre le Buffer et la mémoire, avec une taille du paquet variable	107
4.14	Emission à TX=0dBm avec la taille du paquet variable	108
4.15	Comparaison du courant consommé de différentes puissances d'émis- sions	108
4.16	Phase de reception, L=50 bytes, distance RxTx= 4 m, Puissance(Tx)=0 dBm.	109
4.17	Mode Reception avec Taille du paquet variable	110
4.18	Transfert entre le buffer RX et la mémoire	110
4.19	Flash : courant consommé par la mémoire ST-M25P80 - 1 byte. . . .	111
4.20	L'architecture générale du CC2420, [2]	113
4.21	Emission : comparaison modèle et mesures	118
4.22	Transfert de données MCU-TXFIFO	119
4.23	Réception : comparaison modèle et mesures	119
4.24	Transfert de données RXFIFO-MCU : comparaison modèle et mesures	120
4.25	Comparaison des consommations énergétiques - CC2420(TX-RX) et MSP430(MCU).	120

Liste des tableaux

2.1	Comparaison des caractéristiques des protocoles présentés	33
3.1	Taxonomies des protocoles de routage clustering	38
3.2	Caractéristique du module radio	43
3.3	Matrice booléenne d’existence du lien	82
4.1	Description des Low Power Modes (LPM) du MSP430	103
4.2	Msp430 : Courants consommés comparés dans les différents LPMs. .	104
4.3	Comparaison des courants consommés du module radio en mode émission et réception.	109

Introduction Générale

Sommaire

1.1	Motivation	1
1.2	Contributions	3
1.3	Structure de la thèse	4

1.1 Motivation

Grâce aux progrès technologiques réalisés au cours de ces dernières décennies, la société se rapproche rapidement d'une ère de systèmes d'informations ubiquitaires composés de nombreux systèmes numériques distribués dans notre environnement physique. L'idée de base repose sur les outils informatiques embarqués dans des objets de la vie quotidienne. En d'autres termes, c'est une ère de convergence dans laquelle un ensemble d'équipements disparates communique discrètement à travers un tissu de réseaux hétérogènes.

Les systèmes de mesure distribués contribuent à cette évolution à travers les développements des réseaux de capteurs sans fil. La miniaturisation des composants microélectroniques, micromécaniques et de communication a permis la réalisation de noeuds de capteurs communicants sans fil, permettant un déploiement rapide. Le caractère opérationnel et les fonctionnalités attendues, dépendent essentiellement d'une contrainte : l'énergie et sa gestion.

Le Réseau de Capteurs Sans Fil (RCSF) est une technologie qui a un domaine d'applications large : la surveillance des catastrophes naturelles (éruption volcanique, inondations,...), la surveillance de l'environnement (détecter des incendies,...), la protection des infrastructures, la détection industrielle et des diagnostics, la gestion d'énergie, la localisation d'objets etc.

Ce type de réseau est généralement constitué d'un grand nombre de noeuds de capteurs dispersés aléatoirement dans un environnement à surveiller. Ce "champ de captage" définit le terrain d'intérêt pour le phénomène physique étudié. Les noeuds s'y auto-organisent pour former un réseau de communication afin de transmettre l'information sur un support hertzien d'où le nom de Réseau de Capteurs Sans Fil.

Un noeud du réseau est typiquement composé des éléments suivants : l'élément de détection et de mesure, l'élément de traitement et de stockage, l'élément de communication, et une source d'énergie. L'élément de détection et de mesure est composé de capteurs électroniques capables de mesurer certaines grandeurs physiques : les déformations, la température, la luminosité, la pression, l'humidité, etc. Ceci dans l'environnement où les noeuds de capteurs sont déployés. Le composant de traitement est responsable de la collecte et du traitement des données capturées. Le composant de communication sans fil d'un noeud est responsable de l'émission et de la réception des données capturées à partir d'un noeud vers un autre noeud. Il assure le transfert de toutes ces données mesurées et reçues vers un autre noeud ou jusqu'à une destination finale représentée par la station de base.

En raison des contraintes de miniaturisation, les noeuds sont généralement dotés de ressources très limitées en termes de capacité de calcul, d'espace de stockage de données, de débit de transmission et d'énergie. Ces limitations constituent une partie des problématiques de recherche dans le domaine des réseaux de capteurs sans fil. En particulier, la contrainte liée à l'énergie est un problème fondamental. En effet tous les éléments ayant besoin d'énergie pour fonctionner, la maîtrise de la consommation d'énergie d'un noeud reste une problématique majeure pour la maximisation de sa durée de vie.

La ressource énergétique du noeud de capteur est généralement une batterie servant à alimenter tous ses composants. Dans la plupart des applications du réseau de capteur, les noeuds sont déployés dans des zones difficiles d'accès. De ce fait, la durée de vie globale du réseau dépend complètement de celle de la batterie. L'augmentation de cette durée de vie peut être obtenue par l'utilisation de systèmes de rechargement d'énergie basés sur la récupération d'énergie intermittente (scavenging and harvesting) à partir de l'environnement étudié. Les cellules solaires ou encore les rectennas sont des exemples typiques de ces systèmes. Cependant, la quantité d'énergie qui peut en être extraite est variable et limitée. Par conséquent à ce jour, les réseaux de capteurs sont limités en termes d'énergie disponible.

Les noeuds d'un réseau de capteurs sont déployés dans un environnement afin de surveiller des grandeurs physiques de façon périodique ou événementielle. De façon périodique, il est par exemple possible de mesurer des grandeurs physiques considérés comme des événements statiques. Pour le mode événementiel, nous sommes sur des processus dynamiques. Des seuils des grandeurs physiques sont souvent utilisés comme élément déclencheur de l'activité du noeud. Ils peuvent être par exemple : le mouvement d'un véhicule non désiré dans un champ de bataille, le déplacement des requins dans l'océan, ou un seuil de température dépassé. Dans le cadre de notre étude, nous nous focaliserons sur le cas d'une mesure périodique de grandeurs physiques.

Pour arriver à une efficacité énergétique des RCSFs, et une efficacité en termes de

durée de vie, les organisations géographiques des noeuds doivent être soigneusement envisagées.

Parmi les nombreuses solutions, l'une des méthodes reconnue pour minimiser la consommation d'énergie dans les RCSFs est l'organisation hiérarchique du réseau en clusters. Chaque cluster est sous la gestion d'un noeud maître (le "cluster head" - CH). Dans cette organisation, les noeuds d'un cluster envoient des informations au "cluster-head" correspondant. Le "cluster-head" est le seul habilité à communiquer les informations agrégées directement aux stations de base ou éventuellement à d'autres CHs dans sa portée radio.

1.2 Contributions

Ce projet de thèse vise à analyser et modéliser le comportement énergétique d'un noeud et à optimiser différents protocoles de communication de réseaux de capteurs. L'optimisation énergétique comporte deux aspects : Le premier concerne l'organisation hiérarchique et géographique des noeuds du réseau. Nous avons fait le choix de concentrer notre étude sur les méthodes de clustering liées à la disposition géographique des noeuds. Le second repose sur le protocole de communication. Celui-ci doit assurer un échange d'information à moindre coût énergétique.

Pour atteindre ces deux objectifs, nous concentrons notre étude sur la distribution de l'énergie dans l'ensemble du réseau. Nous proposons une famille de fonctions convexes permettant d'introduire la variance et la moyenne arithmétique d'énergie restant du noeud, afin de développer des méthodes d'estimation et prédiction de son état futur. Les sollicitations des noeuds du réseau seront ainsi optimisées et mieux distribuées.

Il est intéressant d'analyser ces fonctions sur les deux approches traditionnellement utilisées dans le cadre des méthodes de clustering à savoir, l'approche centralisée et l'approche distribuée. Pour l'approche centralisée, il n'y a pas de difficulté vue que l'hypothèse de l'algorithme de base est itérative, c'est à dire que tous les calculs se font à la station de base, laquelle n'a pas de contraintes de ressources. Dans le cas de l'approche distribuée, les calculs se font localement. Afin de prendre en compte les contraintes au niveau mémoire et capacité de calcul, il est proposé deux méthodes de calcul pour exprimer la variance et la moyenne d'énergie résiduelles du noeud pour éviter sa surcharge.

L'une de ces deux méthodes a été déduite de l'expression de la durée de vie du réseau et de la consommation globale du réseau. L'autre expression est inspirée de la moyenne glissante récursive de l'énergie restante. En considérant les résultats obtenus, un algorithme distribué est conçu à partir de l'algorithme centralisé. Ces différents modèles ont été évalués pour différents algorithmes de protocole (centralisé ou distribué) à optimiser.

Par ailleurs, la prise en compte de la surconsommation énergétique des "cluster-heads" due à la transmission directe vers la station de base a permis d'envisager une extension de ces méthodes à travers deux propositions de protocoles multisaut. L'objectif principal est de permettre le passage à l'échelle en construisant des chemins élémentaires entre les "cluster-heads" et la station de base. De plus, l'un de ces protocoles forme une hiérarchie des "cluster-heads" pendant la phase préliminaire de formation des clusters en mettant un accent sur l'auto-ordonnancement et l'auto-organisation pour rendre les réseaux le plus adaptable à un grand nombre de noeuds.

L'injection de cette fonction dans des protocoles existants permet de conclure que cette méthode génère une durée de vie importante du noeud et par conséquent du réseau entier.

Une autre partie importante de ce travail de thèse porte sur l'énoncé d'un nouveau modèle énergétique du noeud de réseau TelosB utilisé dans le cadre de ce travail. L'approche de la consommation suit une structure "bottom-up". A savoir l'identification de fonctions électroniques typiques importantes assurant un bon fonctionnement du noeud et du réseau. Ce point permet d'avoir la modélisation énergétique plus détaillée d'un noeud de réseau, dans son mode de fonctionnement de base.

La caractérisation énergétique de ces fonctions s'appuie sur le développement d'une instrumentation de mesure de la consommation électrique. La chaîne de mesure est ainsi présentée du capteur au conditionneur, jusqu'au stockage et la représentation des données de mesure. Les résultats des mesures nous permettent d'identifier les corrélations directes entre les différents paramètres de fonctionnement du noeud. Ils permettent aussi de mieux caractériser la consommation électrique selon le séquençement des états d'un noeud. De connaissance des fonctions, de leur séquençement et de leur durée d'activation, il est possible d'établir une modélisation énergétique du noeud en fonction des principaux paramètres identifiés.

1.3 Structure de la thèse

Ce manuscrit est organisé en 5 chapitres. Cette introduction pose la problématique générale de cette thèse notamment la minimisation de la consommation énergétique pour un protocole de communication et la notion de durée de vie du RCSF.

Dans le chapitre 2, nous présenterons brièvement les caractéristiques générales des RCSFs. Ensuite, nous énumérons les intérêts de la technique de clustérisation, et présenterons également les critères pertinents des algorithmes existants dans l'objectif d'étudier les différents aspects d'optimisation et leurs objectifs. Une étude détaillée de quelques approches proposées dans la littérature est aussi abordée.

Dans le chapitre 3, les différents modèles mathématiques de consommation éner-

gétique seront présentés en vue de l'optimisation énergétique des protocoles de communication. Nos différentes propositions en termes de protocole de routage de type hiérarchique seront présentées par la suite. Nous conserverons comme élément de référence en termes de comparaison des propositions la durée de vie des RCSF et le caractère homogène de la distribution de l'énergie résiduelle dans le réseau.

Le chapitre 4 sera dédiée au développement instrumental pour un noeud du réseau. Il a pour but d'évaluer les différentes consommations d'énergie des composants fonctionnels qui interviennent dans le processus de fonctionnement d'un noeud. Nous présenterons aussi les différentes étapes suivies pour leur modélisation. Le modèle global final du noeud sera validé par une comparaison avec les mesures.

Et enfin, les conclusions et les perspectives de ce travail de recherche feront l'objet du chapitre 5.

Etat de l'art des réseaux de capteurs sans fil (RCSF)

Sommaire

2.1	Introduction	7
2.2	Description et caractéristiques du protocole de routage hiérarchique	11
2.2.1	La description du réseau hiérarchique	11
2.2.2	Les caractéristiques du réseau hiérarchique	12
2.3	Les algorithmes de clustering	13
2.3.1	Les algorithmes de construction du cluster	13
2.3.2	Les modes de transmission des données	15
2.4	Présentation des principaux protocoles de routage hiérarchique	17
2.4.1	Protocoles basés sur la construction du cluster	17
2.4.2	Protocoles de routage basés sur la transmission de données	30
2.5	Conclusion	32

2.1 Introduction

Le développement des réseaux de capteurs sans fil (RCSF) s'est accéléré ces dernières années. D'après une étude menée par le cabinet *IDTechEx* en 2012, le marché mondial des RCSF devrait croître de 0.45 à 2 milliards de dollars entre 2012 et 2022 [3]. L'étude souligne également que l'autonomie tant du point de vue énergétique que du fonctionnement de ces réseaux, reste un enjeu principal.

La plupart de ces applications nécessite que le RCSF dispose d'une certaine autonomie : qu'il soit en mesure de fonctionner sans assistance, lorsque les interventions et supervisions humaines sont difficiles voire impossibles [4,5]. L'obtention de cette autonomie s'appuie sur des caractéristiques fonctionnelles et matérielles du RCSF lui permettant de résoudre les problèmes résultant d'un événement ou d'incident.

La conception d'un RCSF s'appuie sur un certain nombre de caractéristiques bien identifiées : une durée de vie importante, une scalabilité et un faible coût de production. La durée de vie du RCSF n'est pas définie de manière unique dans la littérature [6]. Elle est définie selon la fonctionnalité désirée du RCSF. Elle repose souvent soit sur une connectivité minimale (épuisement de l'énergie d'un premier noeud ou d'un pourcentage de noeud du RCSF par exemple), ou sur une qualité de service minimale (taux de couverture spatiale).

La scalabilité ("scalability") désigne la caractéristique d'un RCSF de pouvoir changer d'échelle, d'évoluer et de s'adapter aux conditions de fonctionnement requis et aux événements inattendus.

Enfin, un faible coût de production des noeuds est une priorité pour les RCSF pour être économiquement viables. [4, 5, 7–11].

Concernant la durée de vie du réseau, elle dépend en outre de la quantité d'énergie disponible au niveau des noeuds et des modalités de fonctionnement du réseau. Pourtant, pour faciliter le déploiement, cette ressource repose sur une alimentation autonome. Elle est généralement assurée par une batterie. Pour améliorer l'autonomie, l'une des solutions explorées aujourd'hui par la communauté scientifique, consiste à introduire un dispositif de recharge au niveau des réseaux en fonctionnement. Pour conserver l'aspect nomade potentiel des noeuds, les auteurs proposent les sources d'énergie intermittentes, par exemple les cellules solaires [12–15], source d'énergie RF [16–18], ou d'autres types des sources [19–21]. Ces techniques peuvent améliorer la durée de vie, mais actuellement leur intérêt est limité par la faible quantité d'énergie générée par ces systèmes de recharge [22]. Ils sont réservés à des RCSF à faible consommation.

La durée de vie de la batterie conditionne celle du noeud et influe directement sur le fonctionnement global du réseau [23, 24]. Il est donc important d'optimiser la consommation d'énergie du noeud.

D'une manière générale, les travaux de recherche sur l'optimisation de la consommation d'énergie des RCSF peuvent se résumer en cinq grands groupes d'activités :

- **Electronique** [25–29]
 - Conception de circuits à faible consommation
 - Couplage de différentes sources d'énergie
- **Transmission Point-to-point (couches PHY/MAC)** [30–34]
 - Modulation, codage, ...
 - Réduction des sources de communication inutiles (collisions, réceptions collatérales, surcoût des protocoles, etc.)
- **Organisation du réseau** [4, 5, 35–37]
 - Adressage, identité et localisation du noeud
 - Création et maintien d'une hiérarchie connectée (clusters)
 - Sélection de routes efficaces selon plusieurs critères (routage)

- **Transport des données** [38–42]

- Fiabilité

- Congestion, Sécurité

- **Application** [43–47]

- Compression, agrégation, ...

C'est au niveau de la couche réseau (modèle OSI2, [48]) que l'activité de recherche s'est intensifiée ces dernières années. Parmi les différentes pistes en cours, le protocole de routage est l'une des principales problématiques. L'enjeu est de mettre en place un routage adaptatif ainsi que des protocoles de collecte de données qui soient efficaces au niveau énergétique, afin d'optimiser la durée de vie du RCSF. Nous allons nous focaliser plus particulièrement sur les RCSF statiques dans cette étude bibliographique.

Dans les RCSF, comparativement aux réseaux ad-hoc traditionnels, le routage présente encore plus de contraintes de part leurs caractéristiques intrinsèques. On peut dresser une liste non-exhaustive de ces principales contraintes :

- Premièrement, la contrainte de la source énergétique impose des capacités de traitement et de mémorisation ainsi qu'une puissance de transmission limitées.
- Deuxièmement, il est difficile de concevoir et de maintenir le schéma d'adressage et la connaissance de la disposition géographique globale. Souvent, cela demande un mécanisme d'échange d'information et de mise à jour d'adresse périodiquement, ce qui représente une lourde charge ("overhead") pour les réseaux de grande taille.
- Troisièmement, la collecte de données par de nombreux noeuds de capteurs, s'appuie généralement sur une grande probabilité de redondance de données. Ce point doit être pris en considération par les protocoles de routage (collaboration).
- Quatrièmement, la plupart des applications des RCSF héritent le modèle de communication de "many-to-one".
- Dans les applications affectées au RCSF, où le temps est une contrainte, les transmissions de données doivent être effectuées dans un certain laps de temps. Par conséquent, la latence liée à la transmission de données doit être prise en considération dans ces types d'applications (QoS). Toutefois, la consommation de l'énergie est plus importante que la qualité de service dans la plupart des applications où les noeuds de capteurs sont limités en énergie. En effet, elle conditionnera la durée de vie du réseau. Cela conduit souvent à un compromis entre la qualité de service et l'efficacité énergétique.

Plusieurs techniques de routage ont été développées pour intégrer cette contrainte énergétique dans les RCSF. On peut les classer en trois concepts de routage : le routage plat, le routage géographique et le routage hiérarchique.

Dans le concept de routage plat, tous les noeuds exécutent la même tâche et

ont la même fonctionnalité. La transmission des données est effectuée par saut de proche en proche ("hop-by-hop"), ou en multi-saut ("multi-hop"). Elle est basée dans les méthodes de diffusion telles que le "flooding" ou "gossiping", sur lesquelles s'appuient les protocoles tels que SPIN("Sensor Protocol for Information via Negotiation") [49], "Directed Diffusion" (DD) [50], "Rumor" [51], etc. Le routage plat ("flat routing") est relativement efficace pour des réseaux de petites tailles. Cependant, il devient problématique dans les réseaux à grande échelle : le grand nombre de noeuds génère plus de traitement de données au niveau de chaque noeud.

Le routage géographique permet de minimiser l'utilisation des ressources du noeud, mais il nécessite la localisation des noeuds qui sont généralement déployés aléatoirement, notamment "Greedy Perimeter Stateless Routing" (GPSR) [52], "Geographical and Energy Aware Routing" (GEAR) [53], "Energy-Aware Routing" (EAR) [54], "Geographic Random Forwarding" (GeRaF) [37], "A greedy minimum energy consumption forwarding protocol" (GMFP) [55], ... Le développement des techniques d'attribution des données et d'agrégation des données est indispensable pour rendre ce type de routage efficace en énergie.

Enfin, le routage hiérarchique, plus approprié pour les réseaux de grande taille, décompose le réseau en un certain nombre de domaines en limitant le nombre de potentialités d'échange entre chaque domaine. Le routage hiérarchique, associé à une technique efficace de clustering est une technique d'agrégation et/ou de fusion des données, permet d'améliorer la connectivité globale du réseau et l'efficacité énergétique.

Dans la suite de ce chapitre, nous allons nous focaliser sur le routage hiérarchique et analyser les algorithmes les plus marquants ayant été développés pour les RCSF.

Dans un premier temps, la section 2.2 présente brièvement la description du modèle de réseau hiérarchique, ses objectifs et ses avantages.

La section 2.3 décrit les grandes classes d'algorithmes de "clustering" en mettant en exergue les processus de construction du cluster et de transmission des données au sein d'un cluster.

La section 2.4 présente de manière plus détaillée les algorithmes de "clustering" classiques et les plus reconnues, associés au protocole de routage hiérarchique. L'analyse des avantages et inconvénients nous permettra d'identifier les paramètres pertinents pour le travail d'optimisation présenté au chapitre 3.

Enfin, la section 2.5 conclue ce chapitre.

2.2 Description et caractéristiques du protocole de routage hiérarchique

2.2.1 La description du réseau hiérarchique

Il a été démontré que le regroupement des noeuds en clusters permet d'obtenir une bonne efficacité énergétique, de prolonger ainsi la durée de vie du réseau, pour des structures de grande taille [4–6, 35, 56]. Le protocole de routage hiérarchique ainsi que le protocole de collecte de données associé impliquent une organisation en cluster des noeuds. Ce qui conduit à des économies d'énergie significatives grâce à la fusion et l'agrégation des données au niveau du cluster. Dans une architecture de réseau hiérarchique, chaque cluster a un chef ("cluster-head"), et plusieurs noeuds, communément appelés membres. Le "cluster-head", habituellement, effectue les tâches de fusion et d'agrégation des données. Il prend en charge la communication avec les autres clusters.

Le processus de formation des clusters conduit finalement à deux niveaux hiérarchiques où le "cluster-head" forme le niveau supérieur et les noeuds membres constituent le niveau inférieur. Les noeuds membres transmettent leurs données à leur "cluster-head" respectif. Les "cluster-heads" agrègent les données et les transmettent soit directement à la station de base ou soit par l'intermédiaire des autres "cluster-heads". Ce mode de fonctionnement permet de diminuer du nombre total de paquets relayé de part et d'autre et facilite leur transmission.

Toutefois, puisque les "cluster-heads" envoient systématiquement toutes les données à des distances plus éloignées que celles des noeuds membres, il va de soi qu'ils consomment plus d'énergie que les noeuds membres. Une solution courante pour équilibrer cette consommation d'énergie entre tous les noeuds du réseau, consiste à élire périodiquement de nouveaux "cluster-heads". Ainsi le rôle de "cluster-head" est réparti dans le temps à tous les noeuds du réseau. Un exemple typique de communication de données dans un réseau structuré en clusters, en supposant une communication intra-cluster en "single-hop" et une communication inter-cluster en "multi-hop" est illustré sur la figure 2.1. La station de base est le point de traitement de données fournies par les noeuds membres. Elle rend les données accessibles à l'utilisateur final. Elle est généralement considérée comme fixe et située à une distance assez éloignée des différents noeuds. Les "cluster-heads" peuvent être assimilés à des points centraux reliant les noeuds et la station de base. D'une certaine manière, le "cluster-head" est la destination directe et locale pour les noeuds membres, tandis que la station de base est la destination finale pour les "cluster-heads". De plus, cette architecture formée par les noeuds membres, les "cluster-heads" et la station de base peut être répliquée autant de fois que cela serait nécessaire, créant ainsi, si on le souhaite, une hiérarchie de clusters multi-niveaux [48, 57].

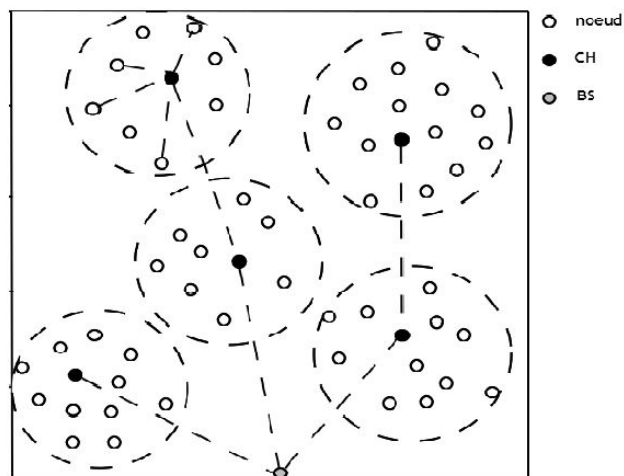


FIGURE 2.1 – communication intra-cluster et inter-cluster dans le réseau

2.2.2 Les caractéristiques du réseau hiérarchique

Comme il a été mentionné précédemment, le clustering hiérarchique dans les réseaux de capteurs peuvent grandement contribuer à l'ensemble du système en terme d'organisation du réseau et d'efficacité énergétique [56]. Par son approche "localisée", le routage hiérarchique est un moyen efficace pour réduire la consommation d'énergie au sein d'un réseau comportant un grand nombre de noeuds.

Au niveau du cluster, l'agrégation et/ou la fusion des données permettent de diminuer le nombre de messages transmis à la station de base [58]. Ce qui la différencie des concepts de routage plat et géographique. En effet, l'obligation de maintenir à jour les tables de routage et de localisation des voisins, peut causer la surcharge des noeuds constituant les routes ou les passerelles avec l'augmentation de la densité des noeuds [59]. Cette surcharge risque de causer des temps de latence dans la communication et des suivis inappropriés des événements.

Le réseau hiérarchique est particulièrement utile pour des applications nécessitant une certaine auto-adaptation pour des centaines ou des milliers de noeuds [5,35]. Ce contexte d'auto-adaptation implique la nécessité d'équilibrer les charges et d'utiliser efficacement les ressources. Outre sa capacité à supporter un changement inattendu au sein du réseau et à diminuer la consommation d'énergie par agrégation de données, le réseau hiérarchique présente de nombreux autres intérêts [5,35]. Il peut localiser les configurations de routage au sein du cluster et réduire la taille de la table de routage stockée au niveau de chaque noeud. Il peut également réduire les trafics (économiser la bande passante) puisqu' elle limite la portée des interactions inter-cluster au niveau "cluster-head", évitant ainsi l'échange de messages redondants entre noeuds. De plus, le réseau hiérarchique peut stabiliser la topologie du réseau

au niveau des capteurs et réduire ainsi l'activité de maintenance de la topologie. Les noeuds membres ne s'occupent alors que de la connexion avec leur "cluster-head" et ne sont pas affectés par les changements au niveau inter-cluster. Par ailleurs, Un "cluster-head" peut aussi planifier les activités du cluster, par exemple de sorte que certains ou tous les noeuds membres peuvent passer en mode en veille. En outre, les noeuds membres peuvent être engagés dans un ordre "round-robin". Leurs temps de transmission et de réception peuvent être déterminés de sorte que la redondance dans la couverture peut être limitée et les collisions au niveau accès media évitées [5, 56].

En résumé, par rapport aux protocoles de routage plat et géographique, le protocole de routage hiérarchique présente une meilleure scalabilité, moins de charge ("less overhead") et moins de consommation d'énergie. Néanmoins le processus de conception des algorithmes de clustering reste cependant un problème délicat. Certaines caractéristiques importantes d'un réseau de capteurs doivent être soigneusement considérées, lors de la mise en oeuvre de l'ensemble des procédures de clustering. Dans de nombreuses applications, ces procédures deviennent très complexes et nécessitent une optimisation particulière [60–62].

2.3 Les algorithmes de clustering

Dans la littérature scientifique [5, 35, 56, 60–62], les méthodes de clustering dans les RCSF s'appuient sur des contraintes qui concernent soit les caractéristiques du cluster, soit les caractéristiques des "cluster-heads" ou encore un mixte des deux. Ces contraintes servent à élaborer les critères à respecter, que l'on devra minimiser lors du partitionnement du RCSF en clusters.

Les algorithmes de clustering des réseaux hiérarchiques comportent deux étapes fondamentales : la partition du réseau selon un critère ou plusieurs critères et l'optimisation des transferts de données. Ces étapes sont effectuées séquentiellement et de manière indépendantes ou couplées spécifiquement selon le contexte [63].

2.3.1 Les algorithmes de construction du cluster

L'objectif de la construction du réseau hiérarchique est d'obtenir un ensemble de clusters qui minimise un critère ou plusieurs critères donnés et de choisir le "cluster-head" au sein de chaque cluster [56]. Ce qui permettra de définir les tables de routage inter-cluster et intra-cluster.

Les méthodes algorithmiques mises en oeuvre pour sélectionner les "cluster-heads" s'appuient sur des paramètres de nature déterministe [6, 64–66] ou adaptative [56, 67–69].

Les paramètres de nature déterministe pris en compte sont souvent les attributs des noeuds de capteurs tels que l'identificateur (ID), l'énergie résiduelle du noeud,

le nombre de voisins ... [6, 64–66]

Pour les paramètres de nature adaptive, on introduit une notion de poids. Ce dernier correspond à une métrique du potentiel du noeud. Ainsi, les "cluster-heads" sont élus parmi les noeuds de capteurs déployés ayant les poids les plus élevés. Ces derniers sont fonction de l'énergie globale du réseau disponible, du coût de la communication, du nombre de sauts, etc. [56, 67–69].

Ces méthodes algorithmiques peuvent s'exécuter de manière aléatoire (ou encore probabiliste) ou itérative (non-probabiliste). Dans les algorithmes à exécution de nature probabiliste, une probabilité a priori est affectée à chaque noeud du réseau de capteurs. Elle est utilisée pour déterminer les rôles des noeuds [?, 57, 67, 68, 70, 71]. En d'autres termes, chaque noeud peut décider de manière indépendante de s'auto-élire en tant que "cluster-head". L'auto-élection permet de faciliter le système de réélection périodique de "cluster-head" et les procédures de réorganisation des clusters. Elle permet également de réagir face aux éventuels changements de topologie réseau, de manière efficace. Ainsi elle assure une rotation appropriée du rôle de "cluster-head" entre les différents noeuds : ce qui augmente l'efficacité énergétique, donc la durée de vie du réseau.

Les algorithmes de clustering à exécution de nature probabiliste permettent généralement d'obtenir des temps d'exécution/convergence rapides et une réduction des paquets échangés pour former le cluster. Ce qui améliore l'efficacité énergétique de cette approche.

Dans l'algorithme de clustering à exécution de nature probabiliste, l'élection et le processus de formation de cluster-head conduit logiquement à des nombres variables de clusters. Ce qui n'est pas le cas dans les algorithmes de clustering à exécution de nature non-probabiliste. Le nombre de "cluster-head", donc des clusters, est prédéterminé.

Ce nombre de clusters est généralement un paramètre critique pour l'efficacité du protocole de routage. Les algorithmes de routage dans les réseaux de capteurs peuvent être exécutés en mode centralisés [56, 72], distribués [56, 70] ou hybrides [68, 73]. Dans les modes centralisés, la station de base (ou un noeud coordinateur) a besoin des informations globales du réseau : topologie, cluster, ... Ils sont dédiés aux applications spécifiques à petite échelle, où la qualité de service et le partitionnement de réseau sont nécessaires. Dans les approches distribuées, un noeud de capteur est capable de devenir "cluster-head" ou d'adhérer à un cluster formé de façon autonome, sans avoir de connaissance sur les caractéristiques globales du réseau ou du cluster.

Les systèmes hybrides mixent les deux approches centralisées et distribuées. Dans ce contexte, les approches distribuées sont utilisées pour l'organisation entre les "cluster-heads" et ses membres. Tandis que les approches centralisées sont souvent utilisées pour former les clusters. Lorsque le processus de clustering est très critique

et complexe, des méthodes d'optimisation d'inspiration biologique, de logique floue, ou d'intelligence artificielle sont utilisées [74–80]. Ces méthodes utilisent des algorithmes itératifs le plus souvent. Il est à noter que dans les algorithmes de clustering à exécution de nature itérative [64, 65, 69, 74, 81–87], des critères plus spécifiques sont utilisés pour l'élection de "cluster-head" et la formation du cluster. Ils sont principalement basés sur la proximité des noeuds (connectivité, degré, ...etc.) et sur les informations reçues des autres noeuds à proximité [64, 65, 69, 81–84]. En plus de la proximité des noeuds, certains algorithmes utilisent une combinaison de plusieurs critères (optimisation multi-critères), notamment l'énergie restante, la puissance de transmission et la mobilité, pour atteindre des objectifs plus globaux [74, 85–87].

La procédure de formation de cluster est typiquement basée sur la communication des noeuds avec leurs voisins (voisins à un saut ou à k sauts). Elle nécessite généralement un échange intensif de messages : ce qui conduit parfois à des périodes de surcharge plus fréquentes que les algorithmes de clustering à exécution de nature probabiliste. En revanche, ces algorithmes sont généralement plus fiables et permettent d'obtenir des clusters robustes et bien équilibrés.

2.3.2 Les modes de transmission des données

Dans certains protocoles de clustering, les noeuds de capteurs transmettent généralement des données au "cluster-head" via une communication directe (communication "one-hop"). Toutefois, la communication intra-cluster "multi-hop" sera requise lorsque la portée de la communication des noeuds est limitée ou que le nombre de noeuds est trop important et que le nombre de "cluster-head" est limité. De plus, si des noeuds de capteurs ne peuvent pas communiquer directement au "cluster-head", leurs données ne seront jamais transmises à la station de base.

Par conséquent, la garantie de la connectivité est un objectif essentiel des protocoles de routage clustering dans les RCSF [35, 88], pour assurer la transmission de données.

Un exemple important est donné par [89] : lorsque certaines informations concernant tous les noeuds de capteurs sont nécessaires pour les protocoles de routage de clustering. Généralement, la communication "multi-hop" dans le routage clustering est surtout utilisée pour la communication inter-cluster pour transmettre les données agrégées à la station de base. Les données transmises par chaque "cluster-head" relais comprennent à la fois ses données et les données à relayer des autres "cluster-head".

Les noeuds proches de la station de base auront à transmettre plus de paquets que ceux qui en sont éloignés [90, 91]. Ce qui entraînera un épuisement plus rapide de la ressource énergétique des noeuds les plus proches de la station de base. Et cela générera un trou autour de la station de base et partitionnera l'ensemble du réseau.

Ce qui empêchent prématurément les autres noeuds dans le réseau de communiquer avec la station de base, malgré le fait que de nombreux noeuds disposent encore beaucoup d'énergie en réserve. Ce phénomène est appelé "trou de l'énergie" [59,92].

Par conséquent, des mécanismes d'évitement des trous d'énergie ("energy hole avoidance") doivent être mis en place, pour garantir la transmission des données [59,92]. En particulier, le regroupement inégal ("uneven clustering") est une des méthodes d'équilibrage de charge [84,93,94]. Ce procédé tend à créer un rayon de cluster en fonction de sa position par rapport à la station de base. Il sera plus petit lorsqu'on sera proche de la station de base. Cela permet une plus faible consommation d'énergie dédiée au traitement des données dans les clusters de petite taille. Ce qui permet d'avoir plus d'énergie pour relayer les données des noeuds distants [92]. Cependant, il s'avère complexe de trouver le rayon théorique optimal d'un cluster [94].

Habituellement, on essaie de réduire au minimum la consommation d'énergie pour la communication intra-cluster. Dans les communications inter-cluster, on sélectionne les routes qui s'appuient sur des noeuds disposant de réserves énergétiques importantes. Ce qui permet de prolonger la durée de vie du réseau. En pratique, les découvertes des routes dans les protocoles de routage de type clustering peuvent être regroupées en 3 classes : les proactives, les réactives et les hybrides.

Dans les réseaux proactifs, toutes les routes entre la source et la station de base sont déterminées au préalable et maintenues indépendamment du trafic de données. La transmission d'un message s'effectue à travers un chemin prédéterminé jusqu'à la station de base. En revanche, aucune route n'est prédéterminée dans les réseaux réactifs. Le routage est sélectionné lorsqu'un message doit être délivré de la source à la station de base. Il existe également des approches hybrides qui utilisent une combinaison des deux idées précédentes.

Dans les protocoles de routage des réseaux hiérarchiques, le mode proactif est le plus souvent adopté par rapport au mode réactif. Le protocole de routage APTEEN [86] est une approche hybride appliquée au RCSF. Selon les besoins des utilisateurs, ce protocole permet de régler certains paramètres et de commuter entre les modes proactifs et réactifs pour transmettre les données. Dans cette section et la précédente, nous avons pu voir les méthodes mises en oeuvre dans les réseaux hiérarchiques tant au niveau de la construction et de l'organisation des clusters qu'au niveau des modes de transmission des données. Les objectifs poursuivis de minimiser la consommation d'énergie du RCSF lors de l'étape de clustering et de la phase de transmission des données s'appuient sur certain nombre de techniques et de critères : l'agrégation-fusion de données, l'équilibrage de charge, la garantie de la connectivité, une latence acceptable, une complexité modérée, etc.

La mise en oeuvre des algorithmes de clustering dédiés au RCSF comportent deux étapes. Tout d'abord, la procédure de partitionnement du RCSF qui s'appuie

sur des critères de formation du cluster ou/et des critères de choix des "cluster-head". Ensuite la procédure de choix des routes, fonction des techniques de routage inter-cluster et intra-cluster. Deux modes de communication sont utilisés pour les RCSF : "simple-hop" et "multiple-hop".

Les méthodes de clustering dans les RCSF se différencient par les choix des différents paramètres, modes et techniques au niveau des deux précédentes procédures.

2.4 Présentation des principaux protocoles de routage hiérarchique

Dans cette section, nous présentons une étude comparative des protocoles de routage en cluster les plus importants pour les RCSF. Nous analyserons les algorithmes classiques et les plus reconnus et mettrons en évidence leurs avantages et leurs inconvénients.

2.4.1 Protocoles basés sur la construction du cluster

2.4.1.1 LEACH

"Low-Energy Adaptive Clustering Hierarchy" (LEACH) [56] est une des approches les plus utilisées dans le routage hiérarchique associé au clustering des RCSF : LEACH s'est avérée une source d'inspiration pour de nombreux protocoles de routage pour les RCSF. L'objectif principal de LEACH consiste à sélectionner des noeuds en tant que "cluster-head" (CH) à tour de rôle, de sorte que la consommation d'énergie élevée nécessaire à la communication avec la station de base (BS) soit partagée entre tous les noeuds du réseau. L'algorithme de LEACH comporte plusieurs tours. Et chaque tour est subdivisé en deux phases : la phase "set-up" et la phase "steady-state". Dans la phase "set-up", les clusters sont organisés (distribution des rôles) tandis que dans la phase "steady-state" les données sont envoyées par le CH à la station de base (BS).

Pendant la phase "set-up", chaque noeud peut potentiellement devenir un CH pour le tour en cours. L'algorithme de choix du CH repose sur les caractéristiques antérieures du noeud (le nombre de fois où le noeud a été CH jusque-là) et sur les contraintes fixées au RCSF (un certain pourcentage de CH recommandé pour le réseau). Auxquels s'ajoute une dimension aléatoire, obtenue par l'affectation au noeud d'un nombre aléatoire x entre 0 et 1.

Ce dernier devient CH pour le tour courant si le nombre x est inférieur au seuil de probabilité $T(n)$ suivant :

$$T(n) = \begin{cases} \frac{N}{N - k * (r \bmod \frac{N}{k})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

où N est le nombre total des noeuds, k est le nombre optimal de CH souhaité, r est le tour courant, et G est l'ensemble des noeuds qui n'ont pas été élus CH durant les N/k derniers tours.

Lorsqu'un noeud est élu CH avec succès, il diffuse un message d'annonce à tous les autres noeuds. Selon l'intensité du signal de l'annonce reçue, les noeuds décident du cluster qu'ils rejoindront pour ce tour et envoient un message d'attachement à son CH. Afin de répartir uniformément la charge d'énergie entre les noeuds, une rotation de CH est effectuée à chaque tour pour la génération d'une nouvelle phase d'annonce en se basant sur l'équation (2.1).

Au cours de la phase "steady-state", les noeuds détectent et transmettent les données à la CH. Les CH compressent les données provenant des noeuds rattachés à leur cluster respectif et envoient un paquet agrégé ou fusionné à la station de base (BS) directement. En outre, LEACH utilise un TDMA/CDMA MAC afin de réduire les collisions inter-cluster et intra-cluster. Après un certain temps, qui est déterminé a priori, le réseau revient à la phase "set-up" et procède à une nouvelle élection de CH. LEACH est une approche complètement distribuée et ne nécessite aucune

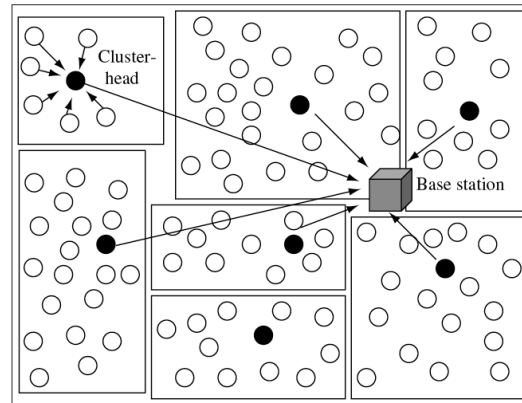


FIGURE 2.2 – Topologie : LEACH Protocol

information globale concernant le réseau. Diverses modifications ont été apportées au protocole LEACH, formant ainsi la famille LEACH, notamment LEACH-C [56], LEACH-DCHS [6], TL-LEACH [57], A-LEACH [95], MR-LEACH [96], LEACH-FL [97], LEACH-R [98], e-LEACH [58], etc. Les avantages de LEACH sont les suivants :

- Tout noeud qui a servi en tant que CH dans un tour précédent ne peut être à nouveau choisi comme CH : ce qui permet d'équilibrer la charge entre tous les noeuds dans une certaine mesure ;
- L'utilisation du système TDMA évite aux CH des collisions ;
- Les noeuds membres du cluster peuvent ouvrir ou fermer les interfaces de communications conformément aux slots qui leur sont alloués pour éviter toute

dissipation d'énergie excessive.

Toutefois, il existe un certain nombre d'inconvénients concernant le protocole LEACH.

- Il effectue une méthode de routage inter-cluster "single-hop", directement du CH à la BS. Or, une communication longue distance directement du CH à la BS peut consommer une grande quantité d'énergie ;
- Malgré le fait qu'une rotation de CH soit effectuée à chaque tour pour réaliser l'équilibrage de charge, LEACH ne peut assurer un véritable équilibrage de charge dans le cas où les noeuds disposent d'une quantité d'énergie initiale différente, car les CH sont élus en termes de probabilité sans considérer leur énergie. Cela pourrait engendrer des déficits et des déséquilibres d'énergie, donc des problèmes de couverture ;
- Dès lors qu'une élection de CH est effectuée en termes de probabilité, il est difficile pour les CH prédéterminés d'être répartis uniformément dans tout le réseau. Ainsi, il peut exister des CH élus qui soient concentrés dans une partie du réseau alors que d'autres noeuds n'ont même pas de CH dans leur voisinage ;
- L'idée de clustering dynamique apporte des charges supplémentaires. Par exemple, lorsque les CH changent, les messages d'annonces sont émis : ce qui diminue le gain en consommation d'énergie.

2.4.1.2 HEED

"Hybrid Energy-Efficient Distributed clustering" (HEED) [68] est un algorithme de clustering "multi-hop" avec prise en considération explicite de l'énergie restante du noeud. De ce fait, HEED présente de meilleures performances en termes d'efficacité énergétique et de durée de vie du réseau. Différent de LEACH, le protocole HEED ne pratique pas le tirage aléatoire des noeuds dans la phase d'élection du "cluster-head". Il effectue l'élection périodique de CHs en se basant sur une combinaison de deux paramètres différents : l'énergie restante du noeud et le coût de la communication intra-cluster. Le premier paramètre permet de sélectionner les noeuds qui possèdent une énergie moyenne restante relativement élevée. Le second permet de bien distribuer les CHs à travers les réseaux.

Initialement HEED fixe un seuil de probabilité : CH_{prob} .

Ensuite chaque noeud du réseau calcule une probabilité CH_{prob} pour devenir CH, par la formule suivante :

$$CH_{prob}(s) = CH_{prob} \times \frac{E_{residual}(s)}{E_{initial}} \quad (2.2)$$

où $E_{residual}(s)$ et $E_{initial}(s)$ représentent respectivement l'énergie résiduelle et initiale du noeud s . La valeur de CH_{prob} , cependant, ne peut pas être inférieure à un certain seuil choisi inversement proportionnel à la valeur $E_{initial}$. Ensuite, chaque

noeud du réseau, après plusieurs itérations, choisit son CH qui minimise le coût de transmission. Si aucun CH n'est entendu, il s'autoproclame CH et propage par la suite sa décision à ses voisins. Pour que l'algorithme converge, chaque noeud double son CH_{prob} à chaque nouvelle itération jusqu'à ce qu'elle atteigne 1. Par conséquent, il existe deux types de statut qu'un noeud de capteur pourrait annoncer à ses voisins. Pour le statut "Tentative status" si $CH_{prob} < 1$, le noeud modifie son état à une itération ultérieure si il constate un moindre coût de CH. Et pour le statut "Final status" si $CH_{prob} = 1$, le noeud devient définitivement un CH. HEED présente quelques avantages comme :

- La formation du cluster décrit dans ce protocole se termine en un nombre d'itérations fini indépendamment du nombre de noeuds dans le réseau.
- L'élection des CHs et la formation des clusters se font de façon complètement distribuée.
- Les clusters obtenus sont bien distribués dans le réseau qui bénéficie de l'utilisation d'énergie restante du noeud.
- La communication "multi-hop" entre les CHs et la BS promeut davantage la conservation d'énergie et la scabilité que la communication directe.

Il existe cependant certaines limites de HEED :

- La formation des clusters à "one-hop" risque de générer un nombre important de clusters si le nombre de noeuds est important. Ainsi, le clustering se rapproche d'une topologie plate de réseau.
- Le mécanisme "Tentative status" CHs favorise l'existence des noeuds isolés (non couverts). Selon l'algorithme du HEED, ces noeuds sont forcés de devenir "cluster-head" et ces CHs forcées peuvent être dans la portée des autres CHs. C'est pourquoi, la communication directe est prévue entre les CHs isolés et la station de base : ce qui oblige les CHs lointains à dépenser plus d'énergie. Il en est de même, lorsque plus de CHs sont générés que le nombre attendu. Cela explique également la consommation d'énergie déséquilibrée dans le réseau [78].
- HEED souffre d'une surcharge conséquente car il a besoin de plusieurs itérations pour former les clusters. A chaque itération, de nombreux paquets sont diffusés. Cette surcharge entraîne une consommation d'énergie significative.

2.4.1.3 DWEHC

"Distributed Weight-based Energy-efficient Hierarchical Clustering Protocol" (DWEHC) [87], est un algorithme de clustering distribué similaire à HEED. L'objectif principal de DWEHC est d'améliorer HEED en construisant des tailles de cluster équilibrées et d'optimiser la topologie intra-cluster en considérant l'emplacement des noeuds. DWEHC et HEED présentent une certaine similitude à savoir, aucune

2.4. Présentation des principaux protocoles de routage hiérarchique 21

hypothèse sur la taille du réseau et de la densité, et la prise en compte de l'énergie résiduelle dans le processus d'élection de CH. Chaque noeud implémente DWEHC individuellement, l'algorithme se termine après plusieurs itérations qui sont mises en oeuvre de manière distribuée. Contrairement à LEACH et HEED, DWEHC crée une structure multi-niveaux pour une communication intra-cluster et limite le nombre d'enfants des noeuds parents. En outre, le seul paramètre poids, calculé localement, est défini pour l'élection de CH dans DWEHC. Après avoir localisé les noeuds voisins dans sa région, chaque noeud calcule son poids en fonction de :

$$W_{weight}(s) = \frac{E_{residuel}(s)}{E_{initail}} \times \sum_u \frac{R - d(u, s)}{R} \quad (2.3)$$

où $E_{residuel}(s)$ et $E_{initail}(s)$ représentent respectivement l'énergie résiduelle et initiale du noeud s , R est la portée du cluster, et d correspond à la distance entre le noeud s et le noeud voisin u . Dans une certaine région, selon l'équation (2.3), le noeud ayant le plus grand poids sera élu CH et les autres noeuds deviendront membres. A ce stade, les noeuds membres sont considérés comme des noeuds de niveau 1 et communiquent directement avec le CH. Un noeud membre peut ajuster progressivement cette adhésion afin d'utiliser le moins d'énergie possible pour atteindre un CH. Compte tenu de la connaissance des noeuds sur la distance les séparant de leurs voisins, ils peuvent évaluer s'il est préférable de rester un membre de niveau 1 ou devenir un membre de niveau h où h est le nombre de sauts le menant au CH. Si un noeud membre peut économiser de l'énergie en atteignant son CH avec plus d'un saut, il deviendra un membre de niveau h . Le processus se poursuit jusqu'à ce que tous les noeuds puissent former la meilleure topologie intra-cluster avec la minimum d'énergie consommée. La consommation d'énergie pour communiquer dans un cluster peut être calculée grâce à la connaissance du noeud concernant sa distance par rapport à ses voisins. Pour limiter le nombre de niveaux, chaque cluster se voit attribuer une portée maximale R à l'intérieur duquel les noeuds membres doivent rester. La structure de cluster multi-niveaux en DWEHC est illustrée sur la figure 2.3.

La communication intra-cluster est effectuée par la méthode TDMA. Chaque noeud parent sonde ses enfants et transmet les données, au fur et à mesure, à son noeud parent jusqu'à ce que les données atteignent le CH. Le noeud parent peut agréger plusieurs paquets de données issus de ses enfants avec ses propres données dans un seul paquet. Pour les communications inter-cluster, les CH recueillent les données provenant de leurs enfants de premier niveau, y compris leurs propres données, et les transmettent à la BS. Les avantages du DWEHC sont :

- Comme HEED, il s'agit d'une méthode de clustering entièrement distribué qui est basée sur la quantité d'énergie du capteur et de la proximité des voisins pour l'élection de CH ;

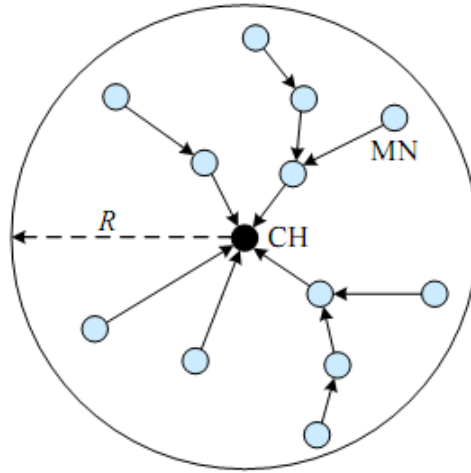


FIGURE 2.3 – Topologie : DWEHC Protocol

- En tenant compte des réserves d'énergie lors de l'élection de CH, DWEHC génère une distribution de CH bien équilibrée et réalise une plus faible consommation d'énergie dans les routages intra-cluster et inter-cluster que HEED ;
- Le processus de clustering du DWEHC se termine après quelques itérations, et ne dépend ni de la taille ni de la topologie du réseau.

Certains inconvénients du DWEHC peuvent être résumés comme suit :

- Comme LEACH, une intercommunication " single-hop ", directement du CH à la BS, est effectuée dans DWEHC. Ainsi DWEHC peut consommer une quantité d'énergie importante, et ne peut donc s'appliquer aux réseaux à grande échelle ;
- Dans le processus de formation du cluster, la nature itérative de DWEHC produit plus de messages de contrôle par rapport à d'autres protocoles.

2.4.1.4 TL-LEACH

"Two-Level Hierarchy LEACH" (TL-LEACH) [57], est une extension de l'algorithme de LEACH. TL-LEACH utilise les deux techniques suivantes pour être efficient en termes d'énergie et de latence. Tout d'abord une technique de formation de cluster aléatoire, adaptatif, en auto-configuration et ensuite un contrôle localisé pour le transfert de données. En TL-LEACH, un CH recueille les données auprès des noeuds comme LEACH, mais au lieu de transmettre les données directement vers la station, il utilise une partie des CH qui se trouve entre lui et la BS comme une station de relai. TL-LEACH introduit la notion de hiérarchie à deux niveaux comme le montre la Figure 2.4 : des top CHs appelés "Cluster Head" primaire CH_i , et ceux du deuxième niveau représentés par les "Cluster Head" secondaire CH_{ij}

et les noeuds ordinaires. L'algorithme est composé de quatre phases principales : la phase d'annonce, la phase de configuration du cluster, l'établissement de programme et la transmission de données. Dans la première phase, chaque noeud décide s'il de-

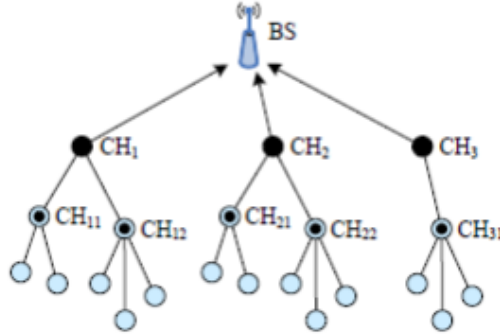


FIGURE 2.4 – Topologie :Two-Level Hierarchy LEACH

vient un CH primaire ou CH secondaire ou encore noeud ordinaire dans chaque tour qui est la même que celle de LEACH. Si un noeud est élu CH primaire, il doit l'annoncer aux autres noeuds. Le mécanisme utilisé dans cette phase est le "Carrier Sense Multiple Access" (CSMA). Par la suite, les noeuds CH secondaires envoient l'annonce aux noeuds ordinaires. Dans cette phase, chaque CH secondaire décide à quel CH primaire, il souhaite s'attacher et il le notifie à ce dernier. De la même manière, chaque noeud ordinaire doit décider à quel CH secondaire il appartient et il l'informe au moyen d'un message dans le sens opposé. Dans la troisième phase, chaque CH primaire crée une table TDMA en attribuant à chaque noeud de son groupe un intervalle de temps (slot) pour transmettre. Chaque CH primaire choisit un code CDMA et informe tous les noeuds de deuxième niveau de son groupe d'utiliser ce code. De la même façon, chaque CH secondaire peut transmettre cette information aux noeuds ordinaires de son groupe en utilisant à la fois ce code et le planning délivré par le CH primaire. Dans la dernière phase, les clusters sont créés et chaque noeud peut transmettre conformément à la table TDMA établie par son CH primaire.

Les avantages de TL-LEACH sont les suivants :

- TL-LEACH utilise une rotation aléatoire des clusters locaux (les CH primaires et secondaires), ce qui apporte une meilleure distribution de charge à travers le réseau ;
- TL-LEACH utilise une coordination localisée, qui apporte de l'évolutivité et de la robustesse dans le réseau ;
- Par rapport à LEACH, le système à deux niveaux de clustering conduit à diminuer la distance de transmission moyenne, et moins de noeuds doivent

transmettre à longue distance, les informations à la BS via TL-LEACH. Cela permet de réduire la consommation d'énergie totale de manière effective.

Cependant, il existe quelques inconvénients à TL-LEACH.

- Bien que la distance moyenne de transmission soit réduite par rapport à LEACH, le routage à deux-sauts inter-cluster de TL-LEACH n'est pas dédié aux réseaux de grande taille, car il utilise seulement deux sauts pour la transmission des données à partir de sources à la BS, ce qui induit une forte consommation d'énergie.
- L'élection de CH sans prise en compte de l'énergie, suppose un réseau homogène idéal. Il ne convient pas pour des réseaux où les états énergétiques initiaux sont hétérogènes.

2.4.1.5 UCS

Le modèle UCS ("Unequal Clustering Size") [99], a été proposé par Soro et Heinzelman pour l'organisation du réseau afin d'équilibrer la consommation d'énergie des CHs, augmentant ainsi la durée de vie du réseau. UCS est le premier modèle de clustering inégal dédié à l'organisation des réseaux WSN. Il se base sur deux principes. Les positions des CHs sont déterminées a priori, et les CHs sont disposés symétriquement selon des cercles concentriques autour de la BS, qui est située au centre du réseau. Il est donc plus facile de contrôler les dimensions réelles des différents clusters. Dans UCS, le champ d'action est supposé circulaire et est divisé

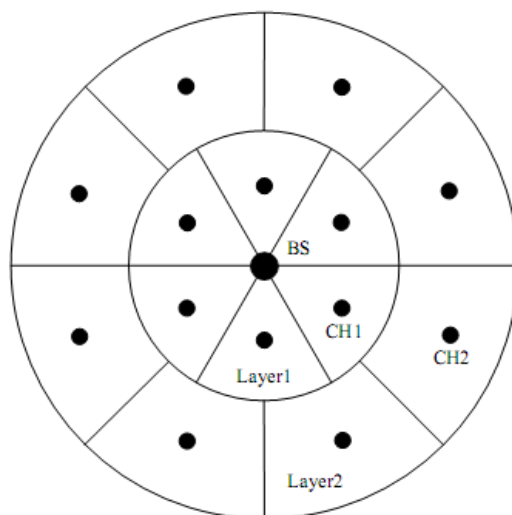


FIGURE 2.5 – Topologie : Unequal Clustering Size

en deux cercles concentriques, dits couches. Afin de simplifier l'analyse théorique, les auteurs assimilent cette zone de détection à un champ en forme de tarte avec

un réseau à couches multiples, représenté sur la figure 2.5. Tous les clusters d'une même couche sont supposés avoir la même taille et la même forme tandis que les dimensions et formes des clusters dans les deux couches sont différentes. La position d'un CH dans la zone délimitant un cluster détermine la consommation d'énergie globale des noeuds appartenant à ce cluster. Afin de minimiser la dissipation de l'énergie totale au sein du cluster, tous les CH doivent se situer au centre de la grappe. Les CH sont déployés de manière déterministe dans le réseau et sont supposés être des super-noeuds, sont beaucoup plus onéreux que les noeuds ordinaires. En faisant varier le rayon de la première couche autour de la BS, tout en supposant un nombre constant de clusters dans chaque couche, la zone couverte par chaque couche peut être changée. Par conséquent, le nombre de noeuds contenus dans un cluster particulier peut être modifié. La transmission de données se fait en plusieurs sauts, où chaque CH choisit de transmettre ses données au CH le plus proche dans la direction de la BS. Les avantages de l'UCS sont :

- En faisant varier le nombre de noeuds dans chaque cluster conformément à la charge de communication prévue, UCS maintient une consommation d'énergie uniforme parmi les CH. Par conséquent, l'énergie totale dissipée pour chaque CH est similaire et UCS peut ainsi prolonger la durée de vie du réseau par rapport au modèle à taille égale ECS ("Equal Clustering Size") ;
- En utilisant le modèle de réseau à deux couches et la méthode de communication inter-cluster à deux sauts, UCS présente une moyenne des distances de transmission plus courte par rapport à LEAH, ce qui réduit considérablement la consommation d'énergie totale.

Cependant, il existe quelques limitations à UCS :

- UCS est limité par l'hypothèse que le réseau est hétérogène, et que les CH sont tout le temps assurés par des super noeuds et déployés à des endroits prédéterminés.
- Les CH doivent être localisés au centre du cluster et l'énergie résiduelle des noeuds n'est donc pas considérée dans UCS comme dans TL-Leach. On constate que la distance moyenne de transmission est réduite par rapport à LEACH.
- Le routage inter-cluster à deux sauts de UCS limite son utilisation à des réseaux de taille réduite.

2.4.1.6 EECS

"Energy Efficient Clustering Scheme" (EECS) [93], est un algorithme de clustering qui convient mieux aux applications de collecte de données périodiques. EECS est un système similaire à LEACH, dans lequel le réseau est divisé en plusieurs clusters et la communication est choisie à un seul saut ("single-hop") entre le CH et la station de base. Dans EECS, les candidats CH se défont pour devenir CH pour un

tour donné. Cette compétition impose aux candidats de diffuser leur énergie résiduelle aux candidats voisins. Si un noeud donné ne trouve pas de noeud ayant plus d'énergie résiduelle que la sienne, il devient un CH. Différent de LEACH quant à la formation de cluster, EECS propose une extension de LEACH par un dimensionnement dynamique des clusters, basé sur la distance du cluster par rapport à la BS. Avec EECS, un noeud choisit le CH de rattachement en considérant non seulement de limiter sa propre consommation d'énergie, mais aussi d'alléger la charge de travail du CH. Ce qui se traduit par la définition de deux facteurs de distance : $d(P_j, CH_i)$ et $d(CH_i, BS)$. Une fonction pondérée $cost(j, i)$ est introduite dans EECS pour un noeud ordinaire P_j lors de sa prise de décision, qui est :

$$cost(i, j) = ((1 - w(P_j))w \times f(P_j, CH_i) + w(P_j) \times g(CH_i)) \quad (2.4)$$

Un noeud P_j , choisit de se rattacher au "Cluster Head" CH_i avec lequel son $cost$ est minimum. Dans l'équation 2.4, f et g sont deux fonctions normalisées respectivement pour les distances $d(P_j, CH_i)$ et $d(CH_i, BS)$:

$$f(P_j, CH_i) = \frac{d(P_j, CH_i)}{d_{f_max}} \quad (2.5)$$

$$g(CH_i) = \frac{d(CH_i, BS) - d_{g_min}}{d_{f_max} - d_{g_min}} \quad (2.6)$$

Où $d_{f_max} = \exp(\max\{d(P_j, CH_i)\})$, $d_{g_max} = \max\{d(CH_i, BS)\}$ et $d_{g_min} = \min\{d(CH_i, BS)\}$. Enfin w est une fonction de P_j tel que :

$$w(P_j) = c + (1 - c) \sqrt{\frac{d(P_j, BS)}{d_{f_max} - d_{g_min}}} \quad (2.7)$$

La fonction f dans $cost$ garantit que les noeuds choisissent le CH le plus proche afin de minimiser les coûts de communication intra-cluster, tandis que la fonction g fait en sorte que les noeuds rejoignent le CH avec une petite distance $d(CH_i, BS)$ pour soulager la charge de travail du CH éloigné de la BS. La fonction w est le facteur de pondération pour le compromis entre f et g . Par ailleurs, la valeur optimale du facteur de pondération c de la fonction w dépend de l'échelle de réseau. Les avantages du EECS peuvent être résumés comme suit :

- Sur la base de l'énergie et de la distance, EECS construit un point d'équilibre entre la consommation d'énergie intra-cluster et la charge de communication inter-cluster ;
- Le clustering est effectué par un dimensionnement dynamique s'appuyant sur la distance du cluster à partir de la BS. De plus, les entêtes des messages s'avèrent plus réduits et l'on assiste aussi à une distribution uniforme des CH par rapport à LEACH.

Les inconvénients du EECS, sont listés ci-après :

- Le choix d'une communication à single-hop dans EECS du CH à la BS peut augmenter considérablement la consommation d'énergie. Par conséquent, il ne convient pas aux réseaux étendus ;
- EECS nécessite plus de connaissance globale sur les distances entre le CH et la BS, et le processus d'agrégation de données globales ajoute plus "d'overhead" pour tous les noeuds ;
- EECS produit beaucoup plus de complexité aux "overhead" (contrôle) parce que tous les noeuds doivent concourir pour devenir CH.

2.4.1.7 EEUC

L'algorithme EEUC ("Energy-Efficient Uneven Clustering") [94, 100], est un algorithme de clustering distribué concurrentiel, où les CHs sont élus par une compétition locale, à la différence de LEACH. Chaque noeud dispose d'un intervalle concurrentiel pré-assigné, qui est d'autant plus petit que l'on se rapproche de la BS. Cela rend EEUC inégal, dans le but d'équilibrer la consommation d'énergie entre CH, tout en résolvant les problèmes critiques. Pendant le processus d'élection des CH dans EEUC, chaque noeud génère un nombre aléatoire. Seuls les noeuds dont le nombre est supérieur à un certain seuil seront activés pour l'élection de CH. La diffusion des messages s'effectue dans un rayon de compétition qui est déterminé par sa distance par rapport à la station de base. Le rayon de compétition d'un noeud s_i est donné par :

$$s_i.R_{comp} = \left[1 - c \frac{d_{max} - d(s_i, BS)}{d_{max} - d_{min}} \right] R_{comp}^0 \quad (2.8)$$

où R_{comp}^0 est le rayon de compétition maximum qui est prédéfini, d_{max} et d_{min} indiquent respectivement la distance maximum et minimum entre les noeuds et la BS, $d(s_i, BS)$ est la distance entre le noeud s_i et la BS, c est un coefficient constant entre 0 et 1. D'après l'équation 2.8, l'intervalle de compétition des noeuds décroît au fur et à mesure que la distance à la BS décroît. Par conséquent, les clusters qui sont proches de la BS sont de tailles plus petites. Ils consommeront donc moins d'énergie durant le traitement des données intra-cluster. Ce qui laissera plus d'énergie pour le relai des trafics inter-cluster. Si un noeud décide de participer à la compétition, basée sur l'énergie résiduelle, il devient un "tentative CH". Ensuite, un "tentative CH" d'une région locale se défie pour devenir un CH réel. Dans EEUC, le routage "multi-hop" est utilisé pour les communications inter-cluster. Les CH choisissent les noeuds-relais pour la transmission de données en fonction de l'énergie résiduelle des noeuds et de la distance à la BS. En d'autres termes, un CH choisira un noeud, disposant de plus d'énergie résiduelle comme son noeud-relai, parmi les deux noeuds ayant les plus faibles coûts de communication par rapport à tous ses voisins. Selon ce qui précède, les avantages de EEUC sont les suivants :

- Pour faire face au problème de "hot spot", EEUC présente un mécanisme de clustering inégal pour équilibrer la consommation d'énergie entre les CH. Par conséquent, ce mécanisme dans EEUC améliore la durée de vie du réseau par rapport à LEACH et HEED ;
- Basé sur le coût de la communication, ce protocole permet d'économiser plus d'énergie par le mécanisme de routage "multi-hop" inter-cluster dans la phase "steady-state", car un CH détermine son noeud-relai parmi les deux noeuds ayant le plus faible coût de communication par rapport à ses voisins.

Cependant, il existe plusieurs inconvénients à EEUC qui sont :

- la création de cluster pour chaque tour impose une surcharge significative, parce que chaque noeud doit diffuser et recevoir une grande quantité de messages (i.e "competitive message") pour l'élection de CH, même si la plupart d'entre eux ne peuvent pas gagner et que la plupart des noeuds élus ne sont pas adaptés pour être CH ;
- Les agrégations de données globales supplémentaires peuvent entraîner plus "d'overhead" pour tous les noeuds et détériorer ainsi les performances du réseau ;
- Le système de routage peut entraîner de nouveaux problèmes de "hot spot", de sorte qu'un seul des deux noeuds ayant les plus faibles coûts de communication parmi les voisins du CH peut être un noeud relai, et ce bien que chacun des deux disposent de peu d'énergie résiduelle.

2.4.1.8 BCDCP

"Base-Station Controlled Dynamic Clustering Protocol" (BCDCP) [72] , est un protocole de routage centralisé avec une BS capable de calcul complexe. L'idée principale de BCDCP est la formation de cluster où chaque CH est associé à un nombre plus ou moins égal de noeuds mobiles pour équilibrer la surcharge de CH et uniformiser la disposition des CH à travers tout le réseau. Au début de la configuration du cluster ("setup phase"), la BS reçoit une information sur l'énergie résiduelle à partir de tous les noeuds du réseau. Sur la base de cette information, la BS calcule d'abord le niveau d'énergie moyen de tous les noeuds du réseau, et choisit ensuite un ensemble de noeuds dont les niveaux d'énergie sont supérieurs à la valeur moyenne. Seuls les noeuds parmi cet ensemble peuvent être élus comme CH pour le tour courant, tandis que ceux à faible énergie peuvent prolonger leur durée de vie en n'effectuant que les tâches d'un noeud ordinaire. Compte tenu de l'ensemble choisi, la BS calcule le nombre de clusters et effectue la tâche de clustering, qui est réalisée par un algorithme itératif de division des clusters. Cet algorithme divise en premier lieu le réseau en deux sub-clusters, et procède ensuite à la division de ces sub-clusters en d'autres petits clusters. Ce processus sera répété jusqu'à ce que le

nombre de clusters souhaité soit atteint. A chaque itération de division des clusters, les deux noeuds qui ont la distance maximale de séparation sont choisis comme CH de l'ensemble. Ensuite, l'ensemble des noeuds restants du cluster en cours de traitement est regroupé au CH dont il sera le plus proche. Après avoir équilibré les deux groupes qui ont approximativement le même nombre de noeuds, les deux sub-clusters sont formés.

Dans BCDP, le système de routage "multi-hop" est adopté pour transférer les données reçues à la BS. Une fois que les clusters et les CH ont été identifiés, la BS choisit la route nécessitant le minimum d'énergie et transfère les informations vers les noeuds sélectionnés avec les détails concernant les clusters ainsi que les CH sélectionnés. Les routes sont sélectionnées en rattachant, premièrement, tous les CH au moyen d'une approche MST ("Minimum Spanning Tree") : ce qui réduit la consommation d'énergie de chaque CH. Puis en choisissant aléatoirement un CH pour transmettre les données à la BS. Par la transmission aléatoire du CH à la BS, la charge transmission est plus ou moins distribuée de manière égale parmi les CH dans BCDP. La figure 2.6 illustre la topologie d'un réseau en BCDP. BCDP

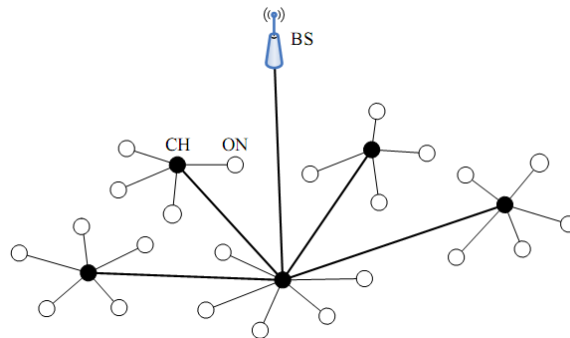


FIGURE 2.6 – Topologie :BCDCP protocol

utilise une BS ayant une quantité d'énergie élevée pour mettre en place les clusters et utiliser MST pour connecter les CH et choisir aléatoirement un CH pour envoyer les données à la BS.

Les avantages de BCDP sont les suivants :

- Les clusters et les routes sont construits par la BS, ainsi BCDP résout le problème de distribution de CH et assure une dissipation de puissance des CH homogène ;
- TDMA est utilisé pour planifier les slots de temps des membres du cluster : ce qui permet aux noeuds d'ouvrir les interfaces de communication dans le cas où les transmissions de données sont nécessaires, réduisant ainsi la consommation d'énergie.

Cependant, il existe quelques inconvénients à BCDP :

- BCDP est un algorithme centralisé qui apporte une évolutivité et robustesse de moindre qualité pour les grands réseaux par rapport aux algorithmes distribués ;
- Chaque noeud doit transmettre des informations concernant leur emplacement et leur niveau d'énergie à la BS au cours du processus de formation de cluster. En conséquence, il augmente la complexité de conception et la consommation d'énergie des noeuds ;
- Compte tenu du système de routage "single-hop", il n'est pas approprié pour les communications à longue distance qui entraînent une forte consommation d'énergie. Ainsi, BCDP n'est pas adapté aux applications avec des réseaux étendus ;
- BCDP n'est pas adapté pour les réseaux réactifs dédiés à la récupération de données périodiques. En effet, les noeuds ne doivent répondre qu'à des événements spécifiques dans les réseaux réactifs.

2.4.2 Protocoles de routage basés sur la transmission de données

2.4.2.1 PEGASIS

"Power-Efficient Gathering in Sensor Information Systems" (PEGASIS) [84], est une amélioration de LEACH. L'idée principale de PEGASIS est pour chaque noeud de communiquer uniquement avec ses voisins les plus proches et de se relayer en qualité de leader pour la transmission. Dans PEGASIS, les emplacements des noeuds sont aléatoires, et chaque noeud a des capacités de détection de données, de communications sans fil, de fusion de données et de calcul de leur position. La charge d'énergie est répartie uniformément entre les noeuds dans le réseau. Dans PEGASIS, les noeuds sont organisés pour former une chaîne, qui peut être soit construite de façon concentrée et diffusée à tous les noeuds, ou soit accomplie par les noeuds eux-mêmes en utilisant l'algorithme glouton ("greedy algorithm"). Si la chaîne est formée par les noeuds eux-mêmes, ils doivent d'abord obtenir les données de localisation de tous les noeuds et calculer localement la chaîne en utilisant le même algorithme. Au cours du processus de formation de chaîne dans PEGASIS, il est supposé que tous les noeuds ont une connaissance globale du réseau et l'on utilise l'algorithme glouton. La construction de la chaîne est initiée à partir du noeud le plus éloigné du noeud "sink" et le noeud voisin le plus proche de ce noeud sera le prochain noeud sur la chaîne. Lorsqu'un noeud sur la chaîne meurt, la chaîne sera reconstruite sur la base des noeuds restants. Pour la collecte des données à partir des noeuds, à chaque tour, chaque noeud reçoit des données d'un voisin. Il fusionne ces données avec les siennes, et les transmet à l'autre voisin sur la chaîne. En passant d'un noeud à un autre, les données fusionnées sont envoyées à "sink" par le "leader" à un emplacement aléatoire sur la chaîne. Le "leader" est important pour les noeuds

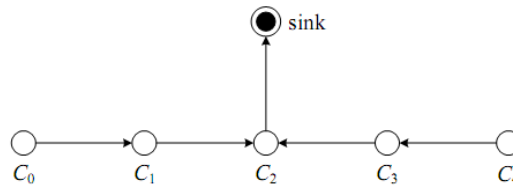


FIGURE 2.7 – Topologie :PEGASIS

pour mourir à des endroits aléatoires.

En effet, l'idée que les nœuds meurent à des endroits aléatoires permet d'améliorer la robustesse du réseau. A chaque tour, une approche de passage de jeton (contrôle) initiée par le leader est utilisée pour démarrer la transmission de données à partir des extrémités de la chaîne. Le système de transmission de données dans PEGASIS est représenté sur la figure 2.7. Sur cette figure, si le nœud C_2 est le leader, il va d'abord passer le jeton le long de la chaîne vers le nœud C_0 . Ensuite, le nœud C_0 passera ses données vers le nœud C_2 . Ensuite le nœud C_2 reçoit des données du nœud C_1 , il va passer le jeton au nœud C_4 , et le nœud C_4 va passer ses données vers le nœud C_2 avec la fusion de données se déroulant tout le long de la chaîne.

PEGASIS présente plusieurs avantages :

- Ce protocole est supérieur à LEACH pour les différentes tailles et topologies de réseau, puisqu'il réduit les "overhead" par rapport à LEACH, et diminue le volume de transmission de données dans la chaîne d'agrégation de données ;
- La charge énergétique est dispersée uniformément dans le réseau. Pour veiller à ce qu'un certain nœud ne soit pas sélectionné en tant que leader et éviter ainsi une mort prématurée de ce nœud, tous les nœuds du RCSF agissent en tant que leader à tour de rôle.

Cependant, il existe quelques inconvénients à PEGASIS :

- Il s'agit de la nécessité d'avoir une vue complète de la topologie du réseau au niveau de chaque nœud pour la construction de la chaîne et ainsi que tous les nœuds doivent être en mesure de transmettre directement au nœud "sink". Ainsi, ce système n'est pas adapté pour les réseaux à topologie évoluant avec le temps ;
- Il est supposé que chaque nœud doit être capable de communiquer avec le nœud "sink" directement, mais typiquement, les nœuds utilisent les communications "multi-hop" avec le nœud "sink" dans les cas pratiques. En outre, une communication à longue distance directement à partir du nœud vers le nœud "sink" risque de consommer une énorme quantité d'énergie ;
- Cette manière de communication souffre de retards excessifs causés par la

chaîne unique pour les noeuds distants et une forte probabilité pour un noeud de devenir un goulot d'étranglement ;

- Il est difficile pour tous les noeuds de maintenir une base de données complète concernant l'emplacement de tous les autres noeuds dans le réseau, de plus, le réseau n'est pas très évolutif puisque tous les noeuds doivent avoir une connaissance globale du réseau et utiliser l'algorithme glouton.

2.5 Conclusion

Dans ce chapitre, nous avons présenté et analysé les algorithmes de routage dédiés aux RCSF. Nous avons vu que l'augmentation de la durée de vie des RCSF reposait sur la diminution d'activité des noeuds. Les réseaux hiérarchiques, par un partitionnement judicieux et le choix d'une communication minimisant la consommation énergétique présentent des avantages indéniables pour les réseaux de grande taille. Nous avons mis en évidence les principales caractéristiques des approches proposées, dans la phase de formation du cluster, et dans la phase de transmission de données.

Dans la dernière partie du chapitre, nous avons présenté plus en détail le fonctionnement de certains algorithmes de routage clustering proposés dans la littérature. Ce qui nous a permis de mieux cerner les métriques et les techniques les plus pertinentes pour choisir les "cluster-heads" ou pour former les clusters.

Le tableau 2.1 synthétise et compare les paramètres et caractéristiques des principaux algorithmes de routage de type clustering que nous avons analysés dans ce chapitre.

Cette étude nous a permis de voir les avantages et inconvénients des principaux algorithmes proposés pour le routage de type clustering. Les approches de clustering présentent les inconvénients classiques des systèmes centralisés avec le concept de "cluster-head". Il est fortement mis à contribution pour la collecte des données et est donc plus susceptible d'être défaillant pour des raisons énergétiques. L'autre point critique est le besoin de décentralisation des décisions, surtout dans les réseaux de grande taille (nombre des noeuds importants) et à grande échelle géographique.

Afin de répondre aux trois principales caractéristiques des RCSF (une durée de vie importante, une scalabilité et un faible coût de production). Nous avons opté pour un algorithme de routage hiérarchique qui possède les caractéristiques suivantes :

- Pas de nécessité de localisation des noeuds (pas de GPS : diminution du coût)
- Robustesse dans le choix des CH avec une méthode probabiliste
- Meilleure connaissance de l'état énergétique du RCSF par la prédiction de l'état futur du noeud.
- Facilité de mise en oeuvre des algorithmes

Protocole	Contraintes pour la formation du cluster								Mise en oeuvre de l'algorithme			
	Caractéristiques clusters				Caractéristiques cluster-head				Procédure de clustérisation			Etablis. Route
	Nombr de cluster	Taille du cluster	Comm. Intra-cluster	Comm. inter-cluster	Existence	Rotation de sélection	Type nœud éligible	Rôle attendu	Mode d'exc. de l'algo	Nature d'exec	Méthode de sélection	Mode de Comm.
LEACH	Variable	Egale	Un-saut	Un-saut	Oui	Péri	Homo	Ag/Fus.	Distribué	Proba	Ad/Ran	Proactive
LEACH-C	Variable	Egale	Un-saut	Un-saut	Oui	Péri	Homo	Ag/Fus.	Centralisé	Proba	Ad/Ran	Proactive
HEED	Variable	Egale	Un-saut	Hybride	Oui	Apéro	Homo	Ag/Fus. Relais	Distribué	Itérative	Adaptive	Proactive
DWEHC	Variable	Egale	Multi-saut	Un-saut	Oui	Apéro	Homo	Ag/Fus. Relais	Distribué	Itérative	Adaptive	Proactive
TL-LEACH	Variable	Egale	Un-saut	2-saut	Oui	Péri	Homo	Ag/Fus. Relais	Distribué	Proba	Adaptive	Proactive
UCS	Variable	Inégale	Un-saut	2-saut	Oui	Péri	Hétéro	Ag/Fus. Relais	Distribué	Proba	Adaptive	Proactive
EECS	Variable	Inégale	Un-saut	Un-saut	Oui	Péri	Homo	Ag/Fus.	Distribué	Proba	Adaptive	Proactive
EEUC	Variable	Inégale	Un-saut	Un-saut	Oui	Péri	Homo	Ag/Fus.	Distribué	Proba	Adaptive	Proactive
BCDCP	Variable	Egale	Un-saut	Un-saut	Oui	Péri	Homo	Ag/Fus.	Centralisé	Itérative	Adaptive	Proactive
PEGASIS	Variable	Egale	Multi-saut	Un-saut	Non	N/A	N/A	N/A	Distribué	Proba	Adaptive	Proactive
TEEN	Fixe	Egale	Un-saut	Multi-saut	Oui	Apéro	Homo	Ag/Fus.	Distribué	Proba	Adaptive	Réactive
APTEEN	Variable	Egale	Un-saut	Multi-saut	Oui	Péri	Homo	Ag/Fus.	Distribué	Proba	Adaptive	Hybride

Etablis. : Établissement, **Comm.** : Communication, **Exc.** : Exécution **Ag/Fus.** : Agrégation/Fusion
Homo/Hétéro : Homogène, Hétérogène **Péri/ Apéro** : Périodique/ Apériodique **Ad/Ran** : Adaptive/Random

TABLE 2.1: Comparaison des caractéristiques des protocoles présentés

- Diminution des communications par agrégation des données
- Répartition uniforme de la charge par une répartition uniforme des CH, une rotation des rôles des noeuds, une communication en "mutli-hop".

Nous aurons comme principaux objectifs, la minimisation du coût énergétique des noeuds en prenant en compte la dispersion de la consommation d'énergie et la connectivité du RCSF. Le RCSF devra garder une bonne distribution des "cluster-head" et un équilibre dans la consommation énergétique des noeuds.

La conception et l'évaluation de ces algorithmes font l'objet du chapitre suivant.

Dans cet état de l'art, nous avons vu le rôle central de la connaissance de l'état énergétique des noeuds. Nous proposons de mieux cerner cette consommation sur un noeud réel au niveau expérimental au chapitre 4. Ce travail aura pour objectif de quantifier cette consommation en fonction d'un certain nombre de paramètres. Ceci afin de déterminer les pistes de travail futures au niveau hardware et aussi améliorer les performances des modèles de consommation des noeuds : *in fine*, améliorer la durée de vie des RCSF.

Optimisation des protocoles de communication

Sommaire

3.1	Introduction	36
3.2	Hypothèses sur le RCSF et définition des métriques	38
3.3	Formulation des problèmes	39
3.3.1	Coefficient de variation	40
3.3.2	Fonction Convexe	41
3.3.3	Description de modèle énergétique du noeud	42
3.3.4	Estimation de consommation d'énergie globale du réseau	43
3.4	Les protocoles centralisés	46
3.4.1	Introduction	46
3.4.2	Limitations de LEACH et LEACH-C protocoles	46
3.4.3	Détermination de nombre de cluster optimal	47
3.4.4	Critère et choix du "cluster-head"	49
3.4.5	Algorithme du k-LEACH et son variant	50
3.4.6	Simulation et Analyse	52
3.4.7	Conclusion	57
3.5	Le protocole distribué : e-LEACH	58
3.5.1	Introduction et Motivation	58
3.5.2	Paramètres de sélection du "cluster-head"	58
3.5.3	Equilibrage de la consommation d'énergie	59
3.5.4	Simulation et Discussion	61
3.5.5	Conclusion	63
3.6	Le protocole décentralisé : LEACH-C2D	65
3.6.1	Introduction et Motivation	65
3.6.2	Sélection du "cluster-head" : Equilibrage d'énergie consommée	65
3.6.3	Limitation de l'équilibrage de la consommation d'énergie	68
3.6.4	Modèle énergétique global exponentiel	70
3.6.5	Résultats : Modèle de consommation globale exponentielle	71

3.6.6 Conclusion	74
3.7 Cluster-Based Multihop Protocols	75
3.7.1 Motivation	75
3.7.2 Forwarding and Routing Stateless Multi-Hop Protocol	76
3.7.3 Distributed Cluster-Based Routing Protocol	78
3.7.4 Simulations et Discussions	85
3.7.5 Conclusion	87
3.8 Conclusion de chapitre	87

3.1 Introduction

Un réseau de capteurs sans fil (RCSF) est un ensemble de noeuds communiquant par des liaisons sans fil, il permet d'observer un phénomène donné dans une zone géographique [101]. Ce type de réseaux présente des applications importantes dans divers domaines avec des technologies récentes [60, 77]. La maîtrise de la consommation d'énergie des RCSF et la maximisation de leur durée de vie restent les problématiques les plus fondamentales. En effet les noeuds sont des composants embarqués de petite taille avec une faible capacité de stockage, de traitement de données et d'énergie [102]. Ils sont alimentés par des batteries dont la capacité est très limitée et qui sont généralement non rechargeables [103]. De ce fait, le principal problème rencontré lors de la conception d'un RCSF est l'autonomie du RCSF. Ainsi, pour répondre à cette problématique, la mise en oeuvre des mécanismes visant à minimiser la consommation énergétique et à garantir une bonne connectivité est importante notamment au niveau des protocoles de communication.

Pour maintenir l'efficacité énergétique des RCSF et disposer d'une durée de vie plus importante, les dispositions géographiques des noeuds doivent respecter un certain nombre de contraintes. La méthode de clustering est considérée comme l'une des meilleures techniques pour organiser le réseau en une hiérarchie connectée [5, 35]. Au niveau du routage, elle permet de minimiser l'utilisation des ressources et d'éviter les diffusions trop fréquentes pouvant surcharger les noeuds du réseau [4, 104].

Nous avons vu que la plupart des protocoles hiérarchiques proposés utilise une communication à un seul-saut (des "cluster-heads" à la station de base) [35, 67, 95]. Cela suppose que les "cluster-heads" peuvent communiquer directement avec la station de base. Bien que cette hypothèse soit appropriée pour un réseau de petite taille, elle pénalise les noeuds éloignés de la station de base [105] compte tenu de la longueur de transmission : ainsi les noeuds "cluster-head" plus éloignés de la station de base meurent plus rapidement. Cela peut entraîner l'existence d'une zone

qui ne sera pas couverte durant le fonctionnement du réseau [106, 107]. Aussi, il est important d'adopter une technique plus efficace pour former les clusters qui assurent le recouvrement du réseau. Et également de choisir judicieusement les noeuds qui prennent le rôle de "cluster-head" afin d'assurer une distribution équilibrée de la consommation énergétique sur l'ensemble des noeuds. Nous choisirons un mode de communication de type "multi-hop" pour les communications inter-clusters, et un mode "one-hop" au niveau local (intra-cluster).

Dans la phase de clustérisation du RCSF, la méthode de sélection du noeud "cluster-head" et de l'affectation des noeuds membres à un cluster est importante. En effet, les noeuds membres détectent l'information au sein de l'environnement dans lequel ils sont déployés et transmettent les données détectées à leur "cluster-head" correspondant. Le noeud "cluster-head" agrège les données détectées par leurs noeuds membres, et les transmet à la station de base. Par conséquent, le noeud "cluster-head" consomme plus d'énergie que les autres noeuds du fait qu'ils exécutent ces tâches supplémentaires. La technique de sélection du "cluster-head" est donc devenue une clé principale dans le processus de la clustérisation. Elle doit assurer un bon équilibrage de la charge et former une bonne distribution des "cluster-heads". La sélection du noeud "cluster-head" parmi les autres noeuds, en choisissant celui qui dispose d'une grande énergie restante, ainsi que le fait de changer le rôle de noeud "cluster-head" périodiquement entre les noeuds, constituent une meilleure technique pour prolonger la durée de vie du noeud réseau.

Nous adoptons lors de la mise en oeuvre des procédures de clustérisation une approche probabiliste, adaptative et distribuée. L'aspect distribué, nous permet de contourner la difficulté d'avoir des informations sur la localisation des noeuds pour assurer la bonne distribution du "cluster-head", mais représente une plus grande difficulté d'implémentation. Le côté adaptatif permet d'intégrer les informations des ressources comme l'énergie restante ou l'énergie dissipée ainsi que l'énergie initiale des noeuds sont introduites pour le choix des "cluster-head" pendant les différents tours de collecte de données. Enfin l'aspect probabiliste dans le choix du "cluster-head", permet une rotation plus importante des noeuds pour ce rôle.

Cependant au vu du caractère probabiliste et distribué, le principal inconvénient est la mauvaise distribution des "clusters-head". Aussi notre travail se focalisera sur ce point. Il améliorera les techniques de choix du "cluster-head" en introduisant une méthode d'estimation et de prédiction de l'état futur du noeud. L'utilisation de la variance et de la moyenne arithmétique d'énergie restante du noeud comme outils de prédiction permettra non seulement d'augmenter la durée de vie du réseau mais aussi d'avoir la bonne distribution du cluster. La technique de clustérisation choisie pour notre RCSF doit être d'une faible complexité, sous peine de surcharge. Elle devra assurer au RCSF une certaine autonomie en terme d'auto-organisation. Pour que le réseau soit stable, il faudra aussi garantir l'existence d'un nombre moyen de

"cluster-heads" à chaque tour.

Avant de passer à la description détaillée des différentes solutions proposées, nous allons tout d'abord présenter les hypothèses et les définitions des métriques utilisées dans la section 3.2.

La section 3.3 présente la formulation des problèmes afin de développer la fonction convexe retenue comme critère, en vue d'optimiser le choix du "cluster-head" et la description de modèle énergétique du noeud permettant d'estimer la consommation d'énergie globale du réseau.

La section 3.4 présente les protocoles centralisés basés sur l'utilisation de cette fonction convexe. Cette dernière est détermininée en fonction de la moyenne et la variance de l'énergie restante du noeud, tandis que la section 3.5 présente le protocole distribué. En s'appuyant sur les résultats de ces deux approches dans la section 3.6, nous présentons une méthode qui nous permet d'aboutir à un protocole distribué (décentralisé) à partir de ce protocole centralisé présenté précédemment.

Dans la section 3.7, pour palier les problèmes de déséquilibre de charge, deux protocoles de routage multi-saut sont présentés. Enfin, nous concluons ce chapitre par la section 4.6.

La table 3.1 suivante synthétise nos choix (en gras) pour les caractéristiques du cluster et du "cluster-head" ainsi que pour la mise en oeuvre algorithmique. La pertinence de ces choix repose sur l'état de l'art mené au chapitre précédent et des caractéristiques que nous avons choisi pour notre RCSF. En s'appuyant sur les méthodes et techniques rencontrées dans la littérature, nous ferons des propositions d'évolution de ces protocoles existants en intégrant nos choix.

Taxonomies de méthodes clusters											
Contraintes Algorithmiques								Mise en œuvre de l'algorithme			
Caractéristique du cluster				Caractéristique du cluster-head				Procédure de clustérisation			Etab. Route
Nb cluster	Uniformité de taille	Com. Intra	Com. Inter	Existe ?	Rotation CH	Type Noeud	Rôle Attendu	Mode d'exécution	Nature d'exécution	Méth. Sélection	Mode Com.
Fixe	Egale	OH	OH	Oui	Péri	Homo	Ag/Fus	Centralisé	Proba	Rand/Adap	Proactive
Variable	Inégale	MH	MH	Non	Apéro.	Hétéro	Relais	Distribué	No-Prob	Déterministe	Réactive
								Hybride	Itérative		Hybride

Nb : Nombre,	Com. Intra ou inter : Communication intra/inter-clusters,	OH, MH : One ou Multi-hop
Ag/Fus : Agrégation/fusion	Péri/ Apéro : Périodique/ Apériodique	Homo/hétéro : Homogène/ hétérogène
Nat. d'exec : Nature d'exécution	Etab. Route : Etablissement de route	Rand/Adap : Random/ Adaptive

TABLE 3.1: Taxonomies des protocoles de routage clustering

3.2 Hypothèses sur le RCSF et définition des métriques

Nous avons considéré un RCSF, constitué d'un ensemble de noeuds distribués de manière uniforme dans un plan à deux dimensions. Le réseau est supposé fixe

avec des caractéristiques homogènes : énergie initiale, capacité de stockage et de traitement de données identiques pour tous les noeuds. Les noeuds capteurs collectent les données et servent une unique station de base. Chaque noeud est capable d'ajuster sa portée de transmission. Une fois déployés, les capteurs sont laissés autonomes. Il est donc impossible de les recharger en cas de déplétion d'énergie ou de panne. Enfin, les capteurs ne connaissent pas leurs emplacements géographiques (absence de système de localisation tel que le GPS).

Pour évaluer notre protocole, nous avons définis un certain nombre de grandeurs utiles à la mise en oeuvre des métriques énergétiques. Nous listons ci-après, les définitions des éléments utilisés pour la mise en oeuvre de nos critères d'optimisation :

- **Rendement ("throughput")** : Ce sont les données que la station de base récolte à chaque tour ;
- **Durée de vie du réseau** : C'est le moment où le RCSF ne remplit plus son cahier des charges initial. On évalue notre protocole selon les (3) métriques suivantes :
 - First Node Died (FND) : Le premier noeud du réseau meurt (le noeud n'a plus d'énergie)
 - Half Node Alive (HAN) : 50% des noeuds du réseau sont encore vivants
 - Last Node Died (LND) : Tous les noeuds du réseau meurent.
- **L'énergie restante** : c'est l'énergie résiduelle après transmission à chaque tour.
- **L'énergie moyenne** : c'est la moyenne arithmétique de l'énergie résiduelle après transmission à chaque tour.
- **L'énergie "variance"** : c'est la variation de l'énergie résiduelle après transmission à chaque tour.
- **Nombre de cluster optimal** : c'est le nombre optimal du cluster-head à chaque tour.

3.3 Formulation des problèmes

La consommation d'énergie d'un noeud dépend fortement de la puissance dissipée pendant les phases de réception, d'émission et de veille. Les puissances consommées en émission/réception sont fonction de la quantité de données échangées, de la distance de transmission, ainsi que des caractéristiques techniques du module radio. L'émission d'un signal est caractérisé par sa puissance, elle même corrélée positivement à sa portée.

Notons que l'énergie de communication représente la portion la plus importante de l'énergie consommée par un noeud capteur [108, 109] .

Le principal objectif des protocoles de routage considérés ici, consiste à maximiser la durée de vie du réseau en maintenant son efficacité. Intuitivement, minimiser

la variance de l'énergie des noeuds permet d'augmenter la durée de vie du réseau. C'est-à-dire, si on arrive à minimiser la variation d'énergie du noeud à chaque tour successif, nous pouvons penser que la consommation énergétique du noeud est stable, et cela pourra entraîner l'équilibrage de charge au niveau du réseau. Cet objectif est atteint si et seulement si :

$$\vartheta_i(t) - \vartheta_i(t-1) \leq \varepsilon \quad (3.1)$$

où $\vartheta_i(t)$ est la variation d'énergie restante du noeud i à chaque tour t , et ε est une constante dont la valeur devra être faible et optimisée.

En considérant le RCSF dans sa globalité, supposons que l'objectif d'optimisation de consommation d'un noeud que l'on s'est fixé auparavant soit atteint. Sachant que l'énergie restante totale est la somme d'énergie restante de tous les noeuds, nous pouvons en déduire que l'énergie restante globale du réseau à chaque tour diminue de manière constante.

Par conséquent, l'énergie globale consommée dans le réseau sera également constante,

$$E_{tot}(1) = E_{tot}(2) = \dots = E_{tot}(t-1) = E_{tot}(t) \quad (3.2)$$

$E_{tot}(t)$: Energie consommée totale à chaque tour t .

Dans la pratique, cet objectif est équivalent au problème d'équilibrage de charge ("energy balance"), qui a pour but de maximiser la durée de vie du réseau en faisant mourir tous les noeuds en même temps. Notons qu'il est parfois préférable de considérer l'énergie restante globale du réseau comme critère.

3.3.1 Coefficient de variation

Pour minimiser la puissance consommée pour la communication, nous devons agir à trois niveaux : minimiser la quantité des données à communiquer et la distance de transmission, utiliser de manière optimale le module radio. Ce dernier est considéré comme figé, aussi on se concentre dans la plupart des cas sur les deux premiers.

Ainsi les métriques reposant sur la distance de transmission et l'énergie restante sont incontournables pour l'élaboration des protocoles de communication hiérarchique du RCSF.

Nous définissons une métrique qui s'appuie sur l'élaboration d'un coefficient de variation (CV) d'énergie restante : par définition le CV est le rapport de l'écart-type $\sigma_i(t)$ à la moyenne $\mu_i(t)$ du noeud i [110, 111].

$$CV_i(t) = \frac{\sigma_i(t)}{\mu_i(t)} \quad (3.3)$$

L'écart type $\sigma_i(t) = \sqrt{\vartheta_i(t)}$ [112] permet de conserver la consommation d'énergie du noeud et prend en compte la dispersion de la consommation d'énergie.

Plus la valeur du coefficient de variation est élevée, plus la dispersion relative autour de la moyenne est grande [113]. Cela veut dire que la distance de transmission précédente est élevée.

Cette métrique permet la comparaison de distributions des valeurs dont les échelles de mesure ne sont pas comparables [114].

Lorsque l'on dispose de valeurs estimées, le *CV* rapporte l'écart-type de l'estimation à la valeur de cette estimation. Plus la valeur du coefficient de variation est faible, plus l'estimation est précise [115–117]. Ce qui est utile pour la prédiction de l'état du noeud.

3.3.2 Fonction Convexe

Dans le processus de sélection du "cluster-head", on envisage d'introduire l'impact du coefficient de variation dans l'expression de calcul du seuil. Nous proposons de pondérer l'expression de calcul du choix des CH par ce coefficient.

A l'état initial, les noeuds sont de même niveau d'énergie, donc l'écart type vaut 0, et la moyenne est égale à la valeur de l'énergie initiale. Ainsi, le coefficient de variation tend vers 0. Pour que notre algorithme converge, nous proposons une fonction convexe qui tient compte de l'objectif principal qui est de maximiser la durée de vie du réseau tout en réduisant la variance de l'énergie restante des noeuds à chaque tour t :

$$\gamma_i(t) = \frac{\alpha \cdot \mu_i(t)}{\alpha \cdot \mu_i(t) + \beta \cdot \sigma_i(t)} \quad (3.4)$$

α et β sont des constantes à fixer.

Donc, nous pouvons modifier le seuil de sélection, comme suit :

$$p'_i(t) = \gamma_i(t) \times p_i(t) \quad (3.5)$$

Avec :

$p'_i(t)$: Nouveau seuil de sélection

$p_i(t)$: Seuil de sélection à optimiser.

Dans l'expression (3.4), à l'état initial, $\gamma_i(t)$ tend vers 1. Cela veut dire que la méthode de sélection n'est pas modifiée par rapport à son expression de départ.

Lors de la progression des activités du réseau, la variance de l'énergie restante des noeuds augmente. A partir de ce moment, $\gamma_i(t)$ joue un rôle de plus en plus important.

Soit $\mu_i(t)$ désigne l'énergie moyenne restante du noeud i , qui peut être déterminée par,

$$\mu_i(t) = \frac{1}{t} \sum_{t=1}^t E_i(t), \quad (3.6)$$

Avec $E_i(t)$ est l'énergie restante du noeud i . Et $\nu_i(t)$ l'énergie "variance" est,

$$\nu_i(t) = \frac{1}{t} \sum_{t=1}^t (E_i(t) - \mu_i(t))^2, \quad (3.7)$$

La consommation moyenne d'énergie par noeud est nécessaire pour calculer la variance.

En effet, tous les noeuds doivent mémoriser leur propre consommation d'énergie à chaque tour. Cela conduit à la surcharge du noeud qui possède des capacités de mémoire et de calcul limitées.

Afin de prendre en compte ces contraintes, on peut développer des méthodes de calcul pour estimer la variance et la moyenne d'énergie restante au niveau du noeud : ceci afin d'éviter la surcharge du noeud. Ces méthodes de calcul s'appuient sur l'existence d'un modèle énergétique du noeud.

3.3.3 Description de modèle énergétique du noeud

Actuellement, plusieurs recherches se focalisent sur la conception des radios à faible consommation vu le rôle central de ces modules dans les modèles de consommation d'énergie [2].

Pour établir ce modèle, nous négligeons la dissipation d'énergie au niveau du module lors de la phase d'émission et de réception. Nous décomposons la consommation d'énergie d'un noeud en fonction des trois activités principales qui lui sont dévolues : la communication, le traitement des données et l'acquisition.

Dans ce travail, on a adopté le modèle du module radio et leurs règles de consommation d'énergie associées (Fig.3.1).

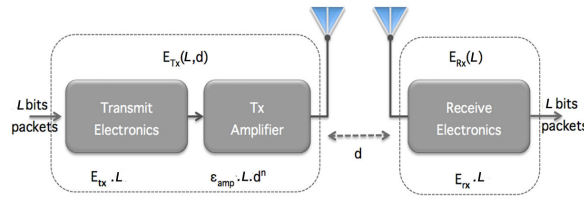


FIGURE 3.1 – Modèle de consommation d'énergie du module radio

- L'énergie consommée pour transmettre un message de taille L [bits] sur une distance entre l'émetteur et le récepteur d [m] est :

$$E_{Tx}(L, d) = \begin{cases} E_{tx} \cdot L + \varepsilon_{fs} \cdot L \cdot d^2, & \text{if } d \leq d_o \\ E_{tx} \cdot L + \varepsilon_{mp} \cdot L \cdot d^4, & \text{if } d > d_o \end{cases} \quad (3.8)$$

- L'énergie consommée pour recevoir un message de taille L [bit] est :

$$E_{Rx}(L) = E_{rx} \cdot L \quad (3.9)$$

Où ε_{fs} et ε_{mp} sont des coefficients d'amplification, dépendant du module de radio choisi. Et d_0 la distance référence qui est égale à la racine carrée du rapport entre ε_{fs} et ε_{mp} .

Pour les noeuds effectuant des traitement de données, une quantité d'énergie supplémentaire sera consommée : c'est le cas des "cluster-heads" effectuant des tâches d'agrégation des données. Cette énergie qui est symbolisée par E_{da} , s'ajoute à notre expression. L'énergie consommée par le traitement des données est calculée en appliquant la formule suivante :

$$E_{da} = 5 \text{ nJ/bit} \quad (3.10)$$

Souvent l'énergie consommée pour effectuer l'acquisition n'est pas très importante devant les autres consommations. Néanmoins, elle varie en fonction du phénomène et du type de surveillance effectuée.

Le tableau 3.2 regroupe les valeurs des caractéristiques radio que nous allons utiliser dans nos simulations.

	Opération	Energie dissipé
E_{tx}	Energie consommée pour transmettre un message	0.208 <i>mJ/message</i>
E_{rx}	Energie consommée pour recevoir un message	0.121 <i>mJ/message</i>
ε_{fs}	Le coefficient d'amplification ε_{fs}	10 <i>pJ/bit/m²</i>
ε_{mp}	Le coefficient d'amplification ε_{mp}	0.0013 <i>pJ/bit/m⁴</i>

TABLE 3.2: Caractéristique du module radio

3.3.4 Estimation de consommation d'énergie globale du réseau

L'algorithme de formation de cluster permet d'assurer qu'il existe k cluster à chaque tour. On considère qu'il existe n noeuds distribués aléatoirement dans une région carrée $M \times M$. Chaque cluster-head dissipe leur énergie, en recevant les données à partir des noeuds membres, en agrégeant les données et en transmettant la moyenne de données agrégées à la station de base.

Par conséquent, l'énergie dissipée du cluster-head au cours d'un tour s'exprime par :

$$E_{CH}(L, d) = \begin{cases} E_{rx} \cdot \left(\frac{n}{k} - 1\right) \cdot L + E_{da} \cdot \frac{n}{k} \cdot L + E_{tx} \cdot L + \varepsilon_{fs} \cdot L \cdot d_{toBS}^2, & \text{if } d \leq d_o \\ E_{rx} \cdot \left(\frac{n}{k} - 1\right) \cdot L + E_{da} \cdot \frac{n}{k} \cdot L + E_{tx} \cdot L + \varepsilon_{mp} \cdot L \cdot d_{toBS}^4, & \text{if } d > d_o \end{cases} \quad (3.11)$$

Où d_{toBS} est la distance entre "cluster-heads" et la station de base. Ainsi, l'énergie dépensée par chaque noeud membre du cluster pour transmettre les données détectées à son "cluster-head" est exprimée par :

$$E_{nonCH} = E_{tx} \cdot L + \varepsilon_{fs} \cdot L \cdot d_{toCH}^2 \quad (3.12)$$

Où d_{toCH} est la distance entre "cluster-head" et leur membres. Et, l'énergie dissipée dans un cluster par tour peut être exprimée par :

$$E_{cluster} = E_{CH} + \left(\frac{n}{k} - 1\right) E_{nonCH} \quad (3.13)$$

Cela nous permet d'en déduire, l'énergie totale dissipée dans le réseau en un seul tour qui peut être exprimée :

$$E_{tour} = k \cdot E_{cluster} \quad (3.14)$$

Finalement, de l'équation (3.11, 3.12, 3.13, 3.14), l'énergie totale dissipée dans le réseau à chaque tour peut être calculée par :

$$E_{tour} = L(n \cdot E_{rx} + n \cdot E_{da} + k \cdot \varepsilon_{mp} \cdot d_{toBS}^4 + n \cdot E_{tx} + n \cdot \varepsilon_{fs} \cdot d_{toCH}^2) \quad (3.15)$$

Le domaine couvert par un cluster-head est modélisé par un cercle de rayon de communication R et de surface $S = \pi R^2$. En effet, si la densité de noeud est distribuée uniformément dans chaque cluster, alors $\rho = k/M^2$ et la superficie totale couverte par les clusters vaut $kS = M^2$. De ces relations, nous en déduisons le rayon du cluster qui peut être déterminé par $R = M/\sqrt{\pi k}$. La distance moyenne entre un "cluster-head" et leur noeud membre est définie par [56, 118] :

$$E[d_{toCH}^2] = \iint (x^2 + y^2) \rho(x, y) dx dy$$

en passant en coordonnée polaire, nous obtenons :

$$\begin{aligned} &= \int_{\theta=0}^{2\pi} \int_{r=0}^R r^2 \rho(r, \theta) dr d\theta \\ &= \rho(r, \theta) \int_{\theta=0}^{2\pi} d\theta \int_{r=0}^{M/\sqrt{\pi k}} r^3 dr = \frac{2\pi k}{M^2} \frac{M^4}{4\pi^2 k^2} \\ &= \frac{M^2}{2\pi k} \end{aligned}$$

D'où :

$$d_{toCH} = \frac{M}{\sqrt{2\pi k}}, \quad (3.16)$$

Finalement,

$$E_{tour} = L(n \cdot E_{rx} + n \cdot E_{da} + k \cdot \varepsilon_{mp} \cdot d_{toBS}^4 + n \cdot E_{tx} + n \cdot \varepsilon_{fs} \cdot \frac{M^2}{2\pi k}) \quad (3.17)$$

En s'appuyant sur ces différentes définitions et modélisations, nous allons mettre en oeuvre les algorithmes d'optimisation des protocoles et évaluer nos différentes propositions au regard de leur efficacité énergétique.

Nous allons présenter tout d'abord les protocoles exécutés en mode centralisé, par souci de simplicité de mise en oeuvre. Puis, nous présenterons les protocoles exécutés en mode distribué pour améliorer les performances des RCSF. A l'issue de ces deux étapes, nous ferons une proposition d'exécution décentralisée qui s'appuie sur les modalités d'exécution centralisée. Et enfin, nous proposerons d'évaluer les performances de protocoles hybrides.

3.4 Les protocoles centralisés

3.4.1 Introduction

Dans les RCSF, la disposition géographique des noeuds du réseau et le choix du protocole de communication sont deux paramètres qui conditionnent fortement sa durée de vie. LEACH et LEACH-C sont considérés comme étant les premiers protocoles de routage hiérarchique [56] qui ont intégré ces paramètres. Ils sont les plus populaires parmi les algorithmes de routage hiérarchiques dédiés aux RCSF. Ils combinent l'efficacité en consommation d'énergie et la qualité de l'accès au média, en se basant sur une organisation du RCSF en cluster et en utilisant les concepts d'agrégation et/ou de fusion de données pour réduire les communications. Ils présentent une meilleure performance en termes de durée de vie et d'excellents résultats en terme d'économie d'énergie.

Dans cette section, nous présentons k-LEACH [119], qui se propose d'améliorer le critère de base dans le choix des cluster-heads dans l'algorithme centralisé LEACH-C.

Afin d'atteindre le critère d'équilibrage de charge qui consiste à répartir la consommation d'énergie et garantir l'obtention d'information sur toutes les zones. Cette proposition utilise une détermination du nombre optimal de cluster-head qui couvre l'ensemble des noeuds dans le domaine du réseau. Les noeuds ayant l'énergie restante moyenne la plus élevée et la plus faible variance de leur consommation énergétique devient "cluster-heads".

3.4.2 Limitations de LEACH et LEACH-C protocoles

Les principaux inconvénients du protocole LEACH sont la distribution inégale des clusters (le choix des Cluster-heads est distribué) et la puissance de transmission élevée requise quand le noeud est situé loin de la base station. Cela entraîne de l'instabilité dans les régions éloignées de la station de base, en raison de la mort prématurée de ses noeuds. Pour pallier ces inconvénients, l'auteur a proposé LEACH-centralisé (LEACH-C) qui intègre l'énergie restante de chaque noeud et l'énergie globale du réseau des ses critères d'optimisation [56].

Au début de chaque tour, la station de base reçoit la localisation actuelle et l'information d'énergie restante à partir de tous les noeuds du réseau. Grâce à l'information sur l'énergie résiduelle, la station de base peut sélectionner les cluster-heads en calculant le rapport entre l'énergie restante du noeud et l'énergie restante globale du réseau de la manière suivante :

$$p_i(r) = \frac{E_i(r)}{E_{tot}}k, \quad (3.18)$$

où $p_i(r)$ est un seuil de probabilité, $E_i(r)$ est l'énergie restante actuelle du noeud i

et k est le nombre de "cluster-heads" et

$$E_{tot} = \sum_{i=1}^N E_i(r), \quad (3.19)$$

En utilisant le seuil de probabilité $p_i(r)$, cela signifie que les noeuds avec une énergie restante plus élevée sont plus susceptibles de devenir cluster-heads que les noeuds avec de faible énergie restante.

Lemma 3.4.1. *$p_i(r)$ est choisie de telle sorte que le nombre prévu de noeuds "cluster-heads" par tour est k .*

Démonstration. S'il existe N noeuds dans le réseau, et $C_i(r) = 1$ si le noeud est éligible comme cluster-head et sinon on lui affecte la valeur zéro.

$$\begin{aligned} E[\#CH] &= \sum_{i=1}^N p_i(r) * C_i(r) \\ &= \sum_{i=1}^N k \frac{E_i(r)}{E_{tot}} * 1 \\ &= \left(\frac{E_1(r)}{E_{tot}} + \dots + \frac{E_N(r)}{E_{tot}} \right) k \\ &= k \end{aligned}$$

Le nombre moyen de cluster-heads par tour dans l'algorithme LEACH-C est k ■

Le nombre de "cluster-head" par tour est un paramètre inconnu du protocole LEACH-C. Comme le choix de "cluster-head" est complètement aléatoire, un risque de mauvaise distribution de "cluster-head" peut survenir [6]. Cela engendrerait rapidement l'apparition d'une zone non couverte.

Dans k-LEACH, la première proposition consiste à déterminer le nombre moyen optimal de "cluster-head" k en minimisant la surface de la zone non couverte. Pour ce faire, on a exprimé le rayon de couverture R du noeud en fonction de la surface de la zone à couvrir dans l'objectif d'améliorer l'organisation du réseau. La deuxième proposition introduit dans la formulation du critère du choix des "cluster-heads", l'expression d'une fonction convexe qui tient compte de l'énergie moyenne restante et sa variance.

3.4.3 Détermination de nombre de cluster optimal

La problématique d'assurer une couverture pour une région donnée, ou un ensemble des points donnés, et de ses variantes, a retenue l'attention dans la communauté des mathématiciens [120–123]. Dans différents articles, des modèles mathématiques ont été proposés pour déterminer le nombre des disques ou de cercles

nécessaires pour couvrir une surface plane. Récemment, ses résultats ont été généralisés dans [124] pour un domaine convexe, et l'auteur a noté que :

- Si un domaine convexe est couvert d'au moins deux cercles isométriques, alors la densité de la couverture par rapport au domaine est plus petite que $\pi/\sqrt{12}$
- Si un domaine convexe est couvert d'au moins deux cercles isométriques, alors la densité de la couverture par rapport au domaine est plus grand que $2\pi/\sqrt{27}$.

Si on note Φ la densité, S est la surface d'un cercle et A est la surface plane. La métrique densité est donc définie par le nombre de disques ou cercles, noté k , par unité de surface. Elle a pour expression :

$$\frac{\pi}{\sqrt{12}} \leq \Phi \leq \frac{2\pi}{\sqrt{27}} \quad (3.20)$$

avec,

$$\Phi = \frac{k * S}{A} \quad (3.21)$$

Pour qu'on puisse utiliser ces deux théorèmes, on peut supposer que les noeuds soient déployés dans une zone carrée (surface plane), ayant pour surface $A = M \times M$. Les noeuds avec un rayon de couverture R peuvent être modélisés par un cercle ou disque de rayon R et de surface $S = \pi R^2$. Et le nombre de cercles k nécessaires pour couvrir une zone n'est autre que le nombre de cluster. Par ailleurs, chaque noeud est capable de détecter des événements à l'intérieur de son rayon de couverture autour d'elle. De toute évidence, pour assurer une couverture reliée à une zone, l'ensemble des cercles ou disques utilisés couvrent tous les points dans cette région, et le graphe de connectivité de toutes les R -cercles doit former une seule composante connexe dans un sens théorique du graphe. Par conséquent, pour qu'un noeud ne soit pas isolé, on suppose que le rayon de communication R_c est $2R$ [125, 126].

On peut en déduire k dans l'équation (3.21), On a :

$$k = \frac{A}{S} \Phi \quad (3.22)$$

En replaçant A et S par ces expressions, on obtient donc,

$$\frac{M^2}{R_k^2 \sqrt{12}} \leq k \leq \frac{2M^2}{R_k^2 \sqrt{27}}, \quad (3.23)$$

où,

$$R_k = \alpha R, \quad (3.24)$$

Avec α : un paramètre d'optimisation permettant de déterminer le nombre optimal de clusters en fonction de l'énergie globale consommée par le réseau. Cette dernière est déterminée par le modèle énergétique du réseau (équation 3.17).

3.4.4 Critère et choix du "cluster-head"

Nous avons vu que le "cluster-head" consomme plus d'énergie que le noeud membre. Il est donc logique de choisir comme cluster-head le noeud ayant le plus d'énergie. L'utilisation de l'énergie restante, dans les critères d'optimisation, permet d'améliorer la durée de vie du réseau, mais elle ne garantit pas la bonne distribution spatiale des "cluster-heads".

Pour résoudre ce problème, LEACH-C requiert les positions des noeuds et utilise un algorithme génétique ([127, 128]) pour déterminer le nombre de clusters optimal k [56].

Notre algorithme k-LEACH présente un algorithme de choix des "cluster-heads" qui tient compte de l'énergie restante et sa variance.

Nous définissons la probabilité de sélection du "cluster-head" de la manière suivante :

$$p'_i(r) = k \frac{\gamma_i(r) * E_i(r)}{E_{tot}} \quad (3.25)$$

Pour garantir que les noeuds ayant le plus d'énergie restante et la plus faible variance deviennent "cluster-head" et que l'on ait k cluster à chaque tour, cette probabilité doit être choisie aléatoirement dans un intervalle I défini comme suit : $[\frac{\sigma_i}{\mu_i + \sigma_i}, 1]$.

Lemma 3.4.2. *L'existence k nombre de "cluster-heads" par tour doit être garanti.*

Démonstration. Détermination du nombre optimal k de "cluster-head".

D'après le lemma (3.4.1), on a :

$$E[\#CH] = \sum_{i=1}^N p_i(r) * C_i(r) = k \quad (3.26)$$

Supposons que $p_i(r)$ est la normalisation de $p'_i(r)$ à 1 dans l'intervalle I . Alors,

$$\begin{aligned} p_i(r) &= \frac{p'_i(r)}{1 - \frac{\sigma_i}{\mu_i + \sigma_i}} \\ &= \frac{p'_i(r)}{\frac{\mu_i}{\mu_i + \sigma_i}} \end{aligned}$$

d'après l'équation (3.5, si l'on choisit $\alpha = \beta = 1$)

$$= \frac{p'_i(r)}{\gamma_i} = p_i(r)$$

En tenant compte de la preuve du lemma (3.4.1). Le nombre prévu de "cluster-head" par tour est donc bien k dans k-LEACH. ■

3.4.5 Algorithme du k-LEACH et son variant

3.4.5.1 Algorithme du k-LEACH

Dans notre protocole, nous avons adopté le concept de la rotation aléatoire du rôle de cluster-head proposé par LEACH-C à chaque tour.

Au début de chaque tour, chaque noeud communique à la station de base (BS) son identification et son énergie résiduelle. Ensuite la station de base calcule la valeur moyenne d'énergie et la variance de tous les noeuds, afin de calculer $p_i(r)$.

Pour que la station de base puisse exécuter cet algorithme d'optimisation, elle doit conserver l'information sur l'énergie restante de tous les noeuds afin de calculer l'énergie restante moyenne et la variance pour tous les noeuds.

Les noeuds dont l'énergie restante est plus importante et la variance est petite seront aptes à devenir "cluster-head" (CH).

Après que la station de base ait reçu les informations de la part des noeuds, l'opération dans k-LEACH peut se résumer en deux phases :

- **Dans la setup-phase :** Choix aléatoire des noeuds CH
 1. BS prend une valeur aléatoire x dans $I \left[\frac{\sigma_i}{\mu_i + \sigma_i}, 1 \right]$.
 2. Si $x < p'_i(r)$, Alors le noeud devient CH durant le tour courant.
 3. Puis, la station de base désigne les "cluster-heads" en envoyant l'ID du noeud.
- **Dans la steady-phase :** Chaque noeud restant décide de choisir le CH le plus proche suivant l'algorithme du LEACH-C.

Dans k-LEACH, la station de base connaît l'énergie et l'identification de chaque noeud, ce qui lui permet de créer un réseau hiérarchique qui dépense moins d'énergie.

Néanmoins la sélection n'est pas autonome et présente une dépense énergétique non négligeable par l'envoi périodique des informations locales de chaque noeud à la station de base au début de chaque phase de sélection.

Or au fil du temps, les noeuds qui ont été "cluster-head" se retrouvent avec un niveau d'énergie plus faible que celui des autres noeuds membres. Néanmoins ils doivent quand même envoyer des informations locales à la station de base au début de chaque tour, ce qui contribue à diminuer leur niveau d'énergie restante. Ainsi, le réseau pourrait devenir déséquilibré en raison d'une grande différence de niveau d'énergie entre les noeuds. Face à ce problème, une version optimisée de k-LEACH est proposée : k-LEACH-VAR.

3.4.5.2 Algorithme du k-LEACH-VAR

Le protocole k-LEACH-VAR conserve les généralités de l'algorithme de k-LEACH. Toutefois, pour améliorer la distribution de la consommation d'énergie, il considère que seuls les noeuds qui ne sont pas "cluster-head" enverront leurs informations

locales à la station de base. Nous supposons qu'à chaque tour, le nombre de cluster est k .

La probabilité de sélection du cluster-head $p_i(r)$ peut être définie comme :

$$p_i(r) = \begin{cases} k \frac{E_i(r)}{\hat{E}_{nCH}}, & C_i(r) = 1 \\ 0, & C_i(r) = 0 \end{cases} \quad (3.27)$$

\hat{E}_{nCH} est l'énergie restante totale du noeud qui n'a pas été "cluster-head". Et $C_i(r) = 0$, si le noeud a déjà été "cluster-head" et $C_i(r) = 1$ s'il est éligible au tour suivant.

Lemma 3.4.3. *Il existe k nombres optimaux de "cluster-heads" à chaque tour.*

Démonstration. L'énergie restante totale du noeud qui n'a pas été cluster-head est définie par,

$$\hat{E}_{nCH} = \sum_{i=1}^N E_i(r) \times C_i(r),$$

Par définition,

$$\begin{aligned} E[\#CH] &= \sum_{i=1}^N p_i(r) * C_i(r) \\ &= k \sum_{i=1}^N \frac{E_i(r)}{\hat{E}_{nCH}} * C_i(r) \\ &= k \left(\frac{\sum_{i=1}^N E_i(r) * C_i(r)}{\hat{E}_{nCH}} \right) \\ &= k \end{aligned}$$

Le nombre optimal cluster-heads est bien k . ■

Dans k-LEACH-VAR, on choisit les "cluster-heads", à l'aide d'une expression qui tient compte de son énergie restante et de sa variance. Ce qui permet d'avoir une bonne distribution du "cluster-head" et de diminuer le déséquilibre des charges. Nous définissons la probabilité de sélection du cluster-head $p'_i(r)$ comme :

$$p'_i(r) = \begin{cases} k \frac{\gamma_i(r) \cdot E_i(r)}{\hat{E}_{nCH}}, & C_i(r) = 1 \\ 0, & C_i(r) = 0 \end{cases} \quad (3.28)$$

Afin de garantir d'avoir k cluster à chaque tour, le cluster-head est le noeud dont la probabilité est inférieure à une valeur choisie aléatoirement dans l'intervalle $I \left[\frac{\sigma_i}{\mu_i + \sigma_i}, 1 \right]$ (même preuve que lemma.3.4.2).

3.4.6 Simulation et Analyse

Dans cette sous-section, nous étudierons les performances de l'algorithme k-LEACH et son évolution nommé k-LEACH-VAR que nous avons proposé. Par la suite nous comparerons ces propositions avec LEACH, LEACH-C et LEACH-SWDN en termes de durée de vie, d'énergie restante globale du réseau ainsi que de la quantité de données délivrée à la station de base.

Nous avons effectué plusieurs simulations en plaçant 100 noeuds, déployés aléatoirement dans un carré de $100 \times 100m^2$, et la station de base est localisée à $x = 50m$ et $y = 0m$. Les noeuds sont dotés d'une quantité d'énergie initiale de $100mJ$. La taille d'un paquet est fixée à 325 bits, dans lesquels il y a 25 bits qui représentent la longueur de l'en tête du paquet. Par ailleurs, nous adoptons le modèle de consommation d'énergie et les caractéristiques du module radio mentionnées dans le tableau (3.2).

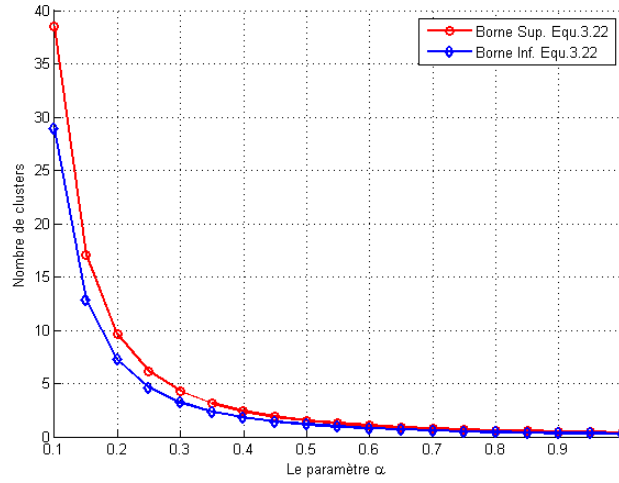
Dans le but d'évaluer ces algorithmes, nous avons déterminé le nombre de clusters optimal qui permet de couvrir le réseau et de minimiser la consommation d'énergie global du réseau. Nous avons effectué des simulations, avec les paramètres qui respectent les relations (3.23) et (3.24), en étudiant l'impact de la valeur du paramètre α sur la dissipation moyenne d'énergie du réseau (équ.3.17).

La figure 3.2.a montre le nombre de clusters créés en fonction du paramètre α lorsque la portée de transmission de chaque noeud est fixé à $R_c = 100m$. Nous remarquons que le nombre de clusters diminue lorsque la valeur de α augmente.

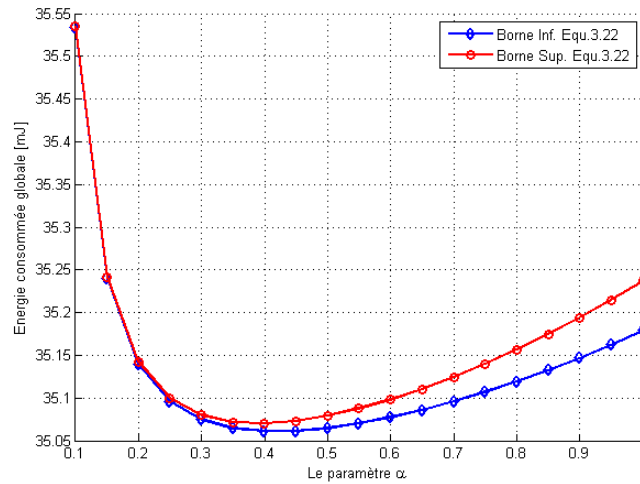
Cela peut s'expliquer par le fait qu'un noeud couvrira un plus grand domaine avec une portée de communication plus étendue si α augmente, entraînant ainsi l'augmentation du nombre de noeuds membres du cluster. Ainsi, le temps qu'un noeud devra attendre pour accéder aux ressources partagées (trame TDMA) va augmenter : ce qui implique que le temps d'attente des "cluster-heads" va augmenter. Cela peut entraîner une surconsommation énergétique de ces dernières.

Par contre, à partir des résultats de la figure 3.2.b, nous pouvons constater que pour des valeurs de α comprise dans l'intervalle $[0.3, 0.5]$ nous avons une consommation d'énergie minimale : en particulier pour $\alpha=0.4$. Ces plages de valeurs permettent de fixer les bornes sous la forme d'un intervalle pour le nombre optimal de clusters. Il est dans l'intervalle $[2, 4]$. On en déduit que la plage de valeur pour le rayon de couverture du cluster est $[30, 50]$. Pour évaluer l'impact de ce rayon de couverture du cluster sur l'efficacité et la durée de vie, nous l'avons choisi dans l'intervalle $[15, 50]$.

Les résultats de la figure 3.3 représentent la durée de vie du réseau en respectant la variation de la portée de couverture du cluster. Nous constatons globalement que lorsque le rayon de couverture du cluster augmente, la durée de vie du réseau tend à augmenter pour les métriques HNA ("Half Node Alive") and LND ("Last Node



(a) Nombre de cluster en fonction du paramètre alpha.

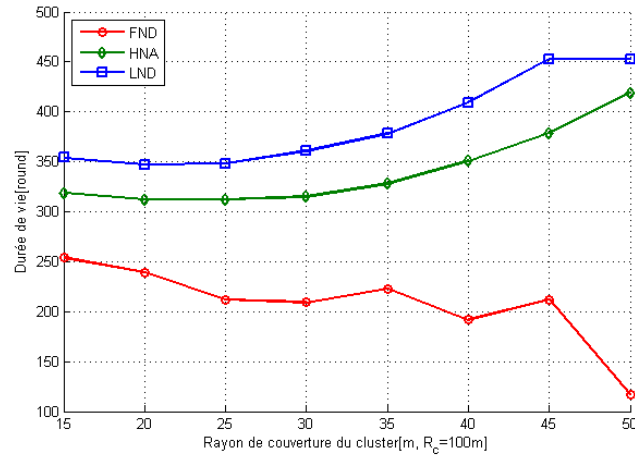


(b) Energie consommée en fonction du paramètre alpha.

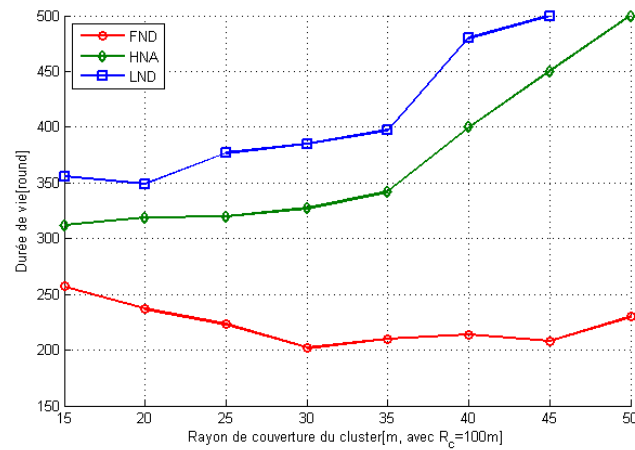
FIGURE 3.2 – Nombre optimal du cluster en fonction du paramètre alpha

Died"), contrairement pour FND ("First Node Died").

Pour la métrique FND, cela peut s'expliquer par le fait que l'augmentation de portée radio d'un noeud réduit le nombre de clusters, conduisant ainsi à augmenter la distance entre les noeuds et leurs "cluster-heads". En effet, dans le cas où les "cluster-heads" choisis sont situés à la frontière des réseaux, les noeuds membres éloignés de leurs "cluster-heads" nécessitent des puissances de transmission très élevées. De plus, la transmission directe de "cluster-head" vers la station de base accélère l'épuisement du noeud le plus éloigné de ce dernier. Pour les métriques de HNA et LND, l'augmentation est remarquable à partir de $R \geq 35$. Cela est dû au fait que



(a) K-LEACH.



(b) K-LEACH variant.

FIGURE 3.3 – Illustration de la durée de vie de k-LEACH et k-LEACH-VAR en fonction de rayon de couverture

la sélection des "cluster-heads" est complètement aléatoire. En effet, compte tenu les résultats de figure 3.2.a, pour le rayon de couverture $R \geq 40$, le nombre moyen du cluster est inférieur ou égal à 2. Cela peut engendrer à un moment l'impossibilité de sélectionner un "cluster-head" lorsque le nombre de noeuds morts augmente au cours du temps. Pour tenir compte de cette éventualité, nous allons évaluer notre proposition en considérant le rayon de couverture $R = 35$.

La figure 3.4 illustre le nombre de noeuds morts au cours du temps (rounds) pour les cinq algorithmes de routage : LEACH, LEACH-C, LEACH-SWDN, k-LEACH et k-LEACH-VAR. Les résultats de cette courbe montrent que k-LEACH-VAR a de meilleures performances que les autres algorithmes. On considère que la me-

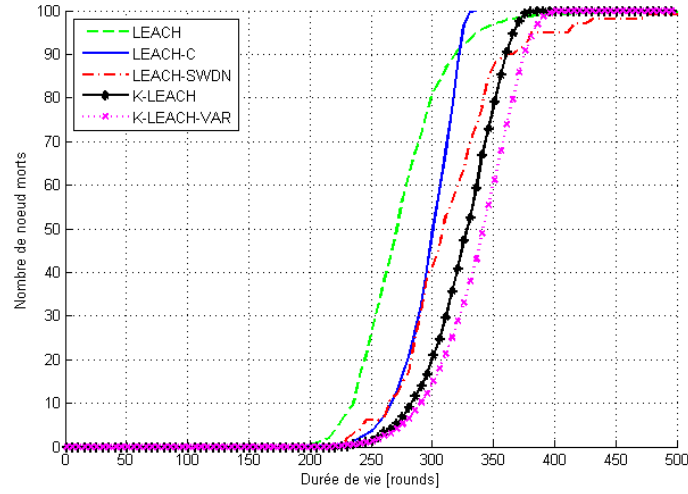


FIGURE 3.4 – Durée de vie du réseau : k-LEACH et K-LEACH-VAR

sure de la durée de vie moyenne est définie par le nombre de tours. Ainsi pour le FND, k-LEACH et k-LEACH-VAR peuvent atteindre environ 254 tours, alors que LEACH, LEACH-C et LEACH-SWDN atteignent 180, 216 et 226 tours respectivement. Pour HNA, k-LEACH-VAR peuvent atteindre environ 330 et 342 tours tandis que LEACH, LEACH-C et LEACH-SWDN atteignent seulement 271, 300 et 306 tours respectivement.

Effectivement, l'algorithme de sélection des cluster-heads du LEACH suppose qu'un noeud peut devenir "cluster-head" avec une même probabilité, ceci peut conduire à une mauvaise distribution des noeuds élus : ce qui entraîne le déséquilibre de la consommation d'énergie du noeud. LEACH-C et LEACH-SWDN améliorent ce point en introduisant la réserve d'énergie dans leurs algorithmes. D'où, de meilleure performance sur la durée de vie par rapport à celle de LEACH. Dans le cas de k-LEACH et k-LEACH-VAR, en plus de l'énergie restante globale du réseau, ils tiennent compte de leur variance afin de bien répartir les noeuds sélectionnés pour devenir "cluster-head". Ainsi, la longévité du réseau augmente. De plus, le rayon de couverture optimale adopté, est déduit de la consommation d'énergie globale du réseau. Cela permet de réduire la distance de communication des noeuds avec leur "cluster-head". Cela permet au noeud d'utiliser leur énergie de manière efficace et favorisant ainsi une augmentation importante de la durée de vie du réseau.

Cette amélioration est encore confirmée par les résultats de la figure 3.5. Cette figure présente l'énergie restante globale du réseau pour les cinq algorithmes au cours du temps (round). Nous voyons que k-LEACH et k-LEACH-VAR conservent plus

d'énergie que LEACH, LEACH-C et LEACH-SWDN : ceci pour les mêmes raisons évoquées précédemment.

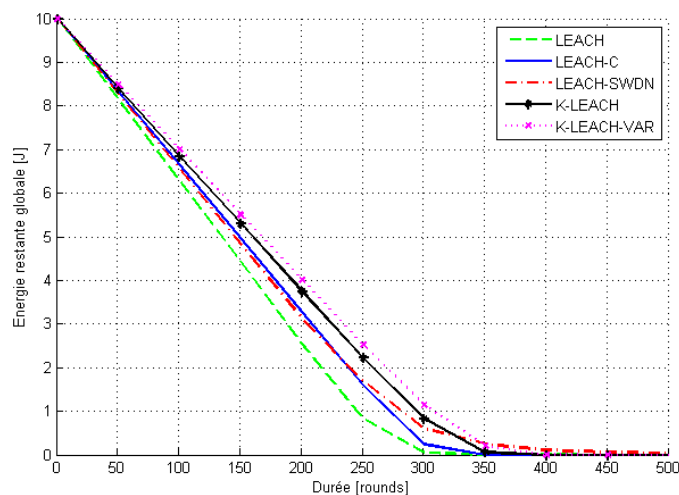


FIGURE 3.5 – Energie restante globale du réseau : k-LEACH et k-LEACH-VAR

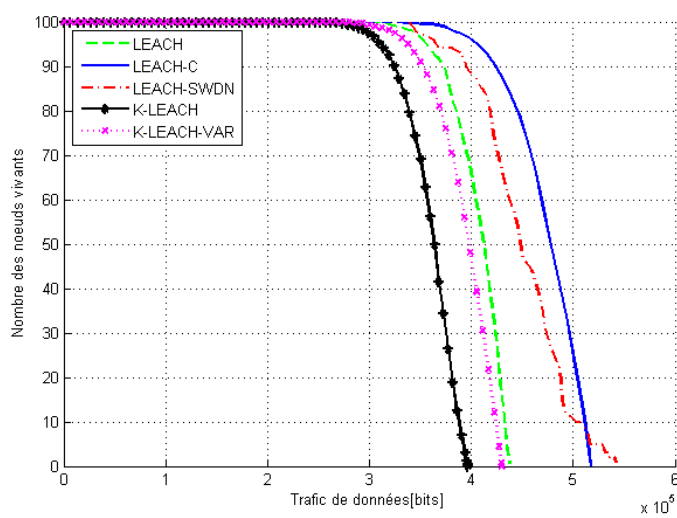


FIGURE 3.6 – Quantités de données reçues par la station de base

Maintenant, nous allons analyser la quantité de données reçues par la station de base pour les cinq protocoles étudiés. La figure 3.6 illustre le nombre de paquets de données fournis à la BS en fonction du nombre des noeuds vivants. Cette quantité est cumulée progressivement au cours du temps. Nos résultats indiquent

également que les quantités de données reçues par la station de base est inférieure à celle des autres algorithmes. Ceci est principalement dû au fait que dans k-LEACH et k-LEACH-VAR, le nombre de cluster-heads dans le réseau est déduit à partir du rayon de couverture du noeud en vue de minimiser la consommation d'énergie. De ce fait, lorsque le rayon de couverture augmente, le nombre de cluster diminue. Or, la quantité de données reçue par la station de base dépend considérablement le nombre de cluster. Par conséquent, le nombre des données générées par la station de base diminue. Cela permet de réduire les pertes dues aux problèmes de communication (interférence,...) en délivrant ainsi la même quantité de données avec moins de consommation d'énergie.

3.4.7 Conclusion

Dans cette section nous avons présenté deux algorithmes de clustering en hiérarchique centralisés qui combinent efficacement différents paramètres notamment l'énergie restante globale du réseau, la variance de l'énergie restante du noeud et la densité de la couverture pour choisir les "cluster-heads" afin de former une bonne distribution de ces derniers. L'avantage principal associé à ce processus de clustering est la détermination du nombre de clusters optimal en fonction de rayon de couverture de cluster : ce qui permet de garantir la fiabilité de données récoltées à la station de base. Les résultats des simulations ont montré que le nombre de clusters créés diminue avec l'augmentation de rayon de couverture. En effet, la valeur optimale du cluster est déduite avec la minimisation de la consommation d'énergie globale du réseau. Celle-ci génère ainsi un nombre réduit de cluster-head. Ce permet à ce dernier de gérer efficacement ses noeuds membres et de garantir un bon fonctionnement du réseau, conduisant à l'augmentation de la durée de vie du réseau. Les résultats des simulations confirment que notre idée de base : minimiser la variance de l'énergie restante de noeuds maximise la durée de vie du réseau.

En effet, pour éviter aussi la surcharge du noeud, les calculs d'énergie restante globale, la variance ainsi que la moyenne d'énergie restante de chaque noeud se font au niveau de la station de base. De ce fait, tous les noeuds doivent transmettre directement leur niveau d'énergie à la station de base. En conséquence, cet étape va générer une augmentation de trafic de contrôle au sein du réseau : ce qui peut générer de la latence. Pour remédier à ce problème, nous proposons une conception de protocole de type distribué dans la section suivante.

3.5 Le protocole distribué : e-LEACH

3.5.1 Introduction et Motivation

Après avoir proposé un algorithme de clustering centralisé dans la section précédente, nous proposons dans cette section un nouvel algorithme de routage hiérarchique distribué, dénoté e-LEACH "Energy-efficient clustering algorithm based on energy variance for Wireless Sensor Networks" [58]. En effet, l'algorithme centralisé nécessite la connaissance totale des informations locales de tous les noeuds pour construire leur cluster [56]. Cela nécessite donc une communication supplémentaire entre les noeuds et la station de base. Cependant, cet échange de communication périodique peut engendrer l'augmentation du flux d'information. De plus, lorsque le nombre de noeuds augmente, cela génère des interférences entraînant la perturbation du réseau. Il convient donc de limiter les échanges entre les noeuds qui ont pour but de transmettre leurs informations vers la station de base. De ce fait, un algorithme adaptatif et auto-configurable est requis afin de rendre le réseau plus *scalable*.

Le protocole e-LEACH utilise un algorithme distribué, où les noeuds s'auto-élisent pour être cluster-head. e-Leach présente une amélioration de LEACH en introduisant de nouveaux paramètres pour la sélection des cluster-head. En effet, LEACH ne fournit aucune garantie sur la bonne distribution des "cluster-heads". Cela s'explique par le fait que chaque noeud devient régulièrement "cluster-head" avec la même probabilité sans tenir compte de sa réserve d'énergie, provoquant ainsi la mort prématurée des noeuds. Pour remédier à cela, e-LEACH prend en considération l'énergie restante et sa variance.

3.5.2 Paramètres de sélection du "cluster-head"

Dans la perspective de minimiser la consommation d'énergie dans les réseaux, e-LEACH vise un algorithme de clustering qui s'auto-configue périodiquement au cours du temps. L'auto-configuration consiste à la formation de cluster et l'élection du "cluster-head". A la différence des autres algorithmes de clustering citées dans [35, 36], cette étude se focalise sur l'amélioration d'élection du "cluster-head", afin de trouver un protocole efficace en énergie, et bien évidemment, dans le but de minimiser la variation de consommation d'énergie des noeuds pour maximiser la durée de vie du réseau.

Sachant que l'énergie consommée du noeud dépend de la distance de transmission et du nombre de ses voisins pendant qu'il est élu "cluster-head". Seule la métrique variance ou écart-type permet de minimiser la dispersion de cette consommation au tour de la moyenne. Autrement dit, la variance n'est autre que l'image de la position du noeud ainsi que leur nombre de voisins. Ceci est utile pour prédire l'état du noeud lors de son élection en tant que "cluster-head" dans le tour actuel. Pour répartir la

consommation d'énergie, l'élection de "cluster-head" se fait périodiquement et en introduisant la variance et la moyenne d'énergie dissipée.

$$\gamma_i(r) = \frac{\mu_i(r)}{\sqrt{\mu_i^2(r) + \nu_i(r)}}, \quad (3.29)$$

L'expression de seuil pour l'élection du cluster-head peut être modifiée comme suit :

$$p_i(r) = \begin{cases} \frac{p}{1-p*(r \bmod \frac{1}{p})} \frac{\gamma_i(r)*E_i(r)}{E_o} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \quad (3.30)$$

Cela signifie que tous les noeuds ayant le plus d'énergie restante et la plus faible variance deviennent "cluster-head" à chaque tour (tous les tours).

3.5.3 Equilibrage de la consommation d'énergie

Afin que le noeud puisse calculer les différents paramètres clés de seuil de l'élection du cluster-head, en occurrence l'énergie moyenne et la variance, il doit mémoriser tout l'état d'énergie précédente. Or, tout noeud présente des contraintes d'incapacité de mémoire et de calcul. De plus, la mémorisation peut engendrer une surcharge ("overhead") du noeud. Rappelons que l'objectif est de maximiser la durée de vie du noeud du réseau. Si on augmente la durée de vie du réseau à R par exemple, alors les noeuds doivent conserver tous les états précédents jusqu'à $(R-1)$. Par conséquent, le noeud peut être surchargé en mémoire et le nombre d'itérations de calcul va augmenter. Cela générerait alors une latence. Ainsi, il est préférable d'utiliser une estimation de la moyenne et la variance dans le but de réduire le cout de calcul et la mémorisation.

Le protocole e-LEACH est proposé dans le but d'équilibrer la consommation de l'énergie entre les noeuds. Pour faciliter les calculs, nous définissons les deux objectifs suivants :

Définition 3.5.1 (pour le réseau). *La consommation de l'énergie équilibrée est atteinte dans le réseau lorsque les noeuds épuisent simultanément leur énergie.*

Par conséquent, la consommation d'énergie globale du réseau à chaque tour sera invariante :

$$E_{tot}(1) = E_{tot}(2) = \dots = E_{tot}(t-1) = E_{tot}(t)$$

Définition 3.5.2 (pour le noeud). *La consommation de l'énergie stable est atteinte dans le réseau lorsque les noeuds dans le réseau, sauf le "cluster-head", épuisent leur énergie simultanément.*

Lorsque le réseau atteint la consommation d'énergie équilibrée (Déf.3.5.1), la durée de vie du réseau peut être définie par le ratio entre l'énergie initiale totale E_{IniTot} et la consommation énergétique globale E_{tot} du réseau :

$$T = \frac{E_{IniTot}}{E_{tot}} \quad (3.31)$$

En effet, l'énergie restante totale est définie par la sommation d'énergie restante des noeuds dans le réseau, qui peut être exprimée par :

$$\sum_{i=1}^n E_i(r) = E_{IniTot} - r * E_{tot} \quad (3.32)$$

Dans un réseau composé de noeuds ayant la même quantité E_o d'énergie initiale, l'énergie initiale totale du réseau peut être déterminée par :

$$E_{IniTot} = n * E_o \quad (3.33)$$

De (3.31) et (3.32), l'énergie restante totale peut être exprimée comme suit :

$$\sum_{i=1}^n E_i(r) = E_{IniTot} \left(1 - \frac{r}{T}\right) \quad (3.34)$$

Soit $\bar{E}(r)$ représentent l'énergie moyenne au tour r du réseau :

$$\bar{E}(r) = \frac{1}{n} \sum_{i=1}^n E_i(r) = \frac{1}{n} E_{IniTot} \left(1 - \frac{r}{T}\right) \quad (3.35)$$

Cette analyse d'estimation est effectuée dans une perspective d'équilibrage de l'énergie, avec comme objectif : chaque noeud doit consommer la même quantité d'énergie moyenne E_{com} à chaque tour. En effet, en tenant compte l'équation (3.31) et la définition 3.5.2, la durée de vie d'un noeud peut être définie comme :

$$T_i = \frac{E_o}{E_{com}} \quad (3.36)$$

Cela nous permet d'en déduire que la consommation d'énergie moyenne par tour d'un noeud peut être s'exprimée :

$$\mu_i(r) = E_o \left(1 - \frac{r}{2T_i}\right) \quad (3.37)$$

Et E_{com} peut être approchée comme :

$$E_{com} = \frac{E_{tot}}{n} \quad (3.38)$$

Avec, l'énergie totale consommée E_{tot} dans le réseau est donnée par la relation (3.15).

Il est à noter que ces résultats d'analyses sont utilisés en tant qu'énergie consommée moyenne de référence. Cela s'explique du fait que l'objectif de mesurer la variance est d'évaluer l'homogénéité de la distribution de consommation énergétique autour de la moyenne. En effet, la variance peut être approchée par :

$$\nu_i(r) = \frac{E_i^2(r) - \mu_i^2(r)}{r}, \quad (3.39)$$

Cette expression permet de considérer la dispersion de la consommation de l'énergie du noeud autour de la référence.

3.5.4 Simulation et Discussion

Afin d'évaluer les performances d'e-LEACH nous avons effectué plusieurs simulations sous Matlab. On a retenu les mêmes configurations du réseau ainsi que les paramètres de simulation de k-LEACH et k-LEACH-VAR. Par le biais de cette implémentation, nous comparons e-LEACH avec d'autres les algorithmes de LEACH, LEACH-C et sa récente amélioration LEACH-SWDN. Les performances sont évaluées par rapport au nombre de noeuds restants, à la consommation d'énergie globale ainsi qu'à la quantité de données délivrée à la station de base.

La figure 3.7 illustre le nombre de noeud épuisés au cours du temps (round) pour les quatre différents protocoles : LEACH, LEACH-C, LEACH-SWDN et e-LEACH. Les résultats de cette courbe montrent qu'e-LEACH affiche une améliora-

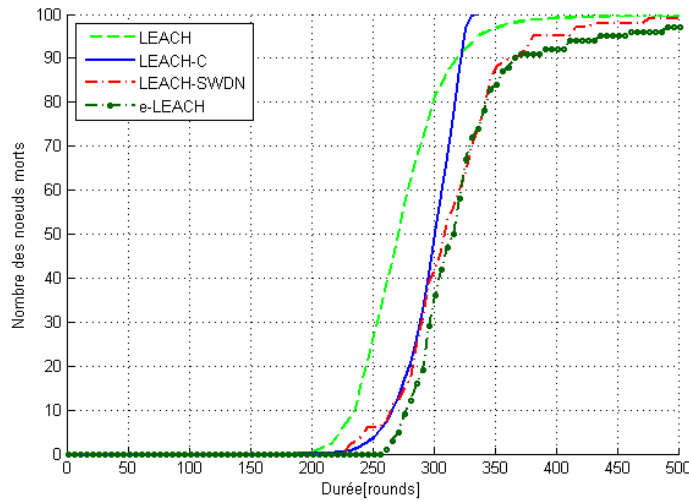


FIGURE 3.7 – Comparaison de la durée de vie d'e-LEACH

tion par rapport aux autres algorithmes. En considérant que la mesure de la durée de vie soit définie par le nombre de tour, si la métrique premier noeud est mort

(FND) est adopté, e-Leach peut atteindre 254 tours, alors que LEACH, LEACH-C et LEACH-SWDN atteignent approximativement que de 200, 220 et 224 tours respectivement. La durée de vie du réseau avec e-Leach est de 27%, 16% et 15% de plus que LEACH, LEACH-C et LEACH-SWDN respectivement. Pour HNA, e-LEACH, LEACH-SWDN, LEACH-C peuvent atteindre 320, 316, 300 tours, respectivement, mais LEACH atteint seulement 250 tours. La durée de vie est améliorée de 28% pour l'e-LEACH par rapport à LEACH.

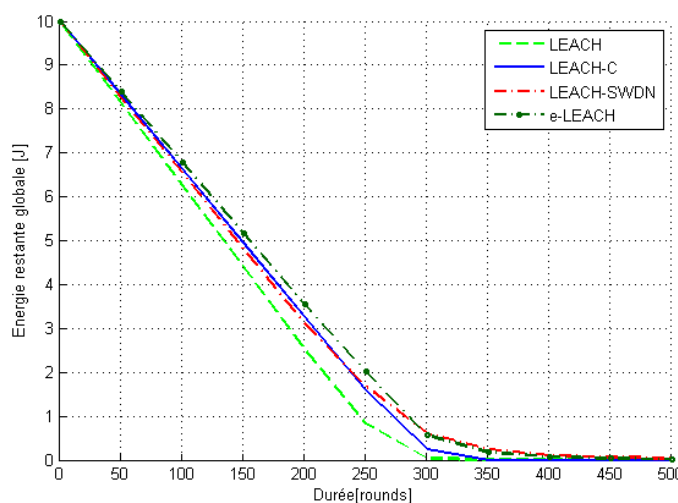


FIGURE 3.8 – Comparaison d'énergie restante d'e-LEACH

Cette amélioration apportée par e-LEACH est encore confirmée par les résultats de la figure 3.8. En effet, cette figure présente l'énergie moyenne restante globale en fonction du temps. Nous voyons clairement que e-LEACH optimise mieux la dépense d'énergie que LEACH, LEACH-C et LEACH-SWDN. Cette optimisation est très importante par rapport à LEACH. Cela vient du fait que dans LEACH, tous les noeuds ont la même probabilité de devenir "cluster-head" sans regarder sa réserve d'énergie, cela provoque une consommation déséquilibrée d'énergie ainsi qu'une mauvaise distribution des "cluster-heads". Tandis que LEACH-C et LEACH-SWDN considèrent ce problème. L'amélioration par rapport à ces deux derniers protocoles s'explique par le fait qu'e-LEACH introduit la variance et l'énergie restante du noeud dans le processus de sélection du "cluster-head". Effectivement, la variation de consommation d'énergie pour un "cluster-head" dépend la distance par rapport à la station de base ainsi que le nombre des membres du cluster. De plus, la variance est déterminée à partir de la consommation moyenne référence déduite de la technique d'équilibrage énergétique du noeud. En effet, e-LEACH utilise la variance de la consommation d'énergie pour rendre le "cluster-head" bien distribué. Ainsi,

la consommation d'énergie est réduite, ce qui se traduit par l'augmentation de la durée de vie du réseau.

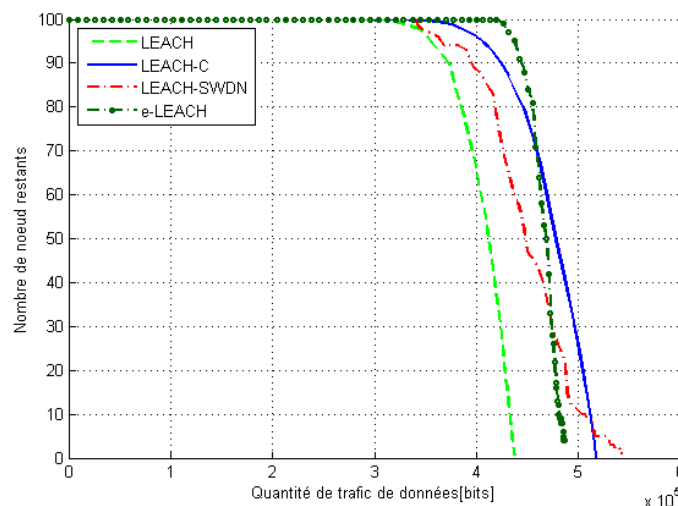


FIGURE 3.9 – Comparaison des traffics au sein du réseau

La figure 3.9 montre le nombre de paquet de données reçues par la station de base en fonction du nombre de nœuds survivants dans le réseau. Les résultats de cette figure illustrent l'efficacité d'e-LEACH en livrant un nombre plus élevé de messages de données comparativement aux protocoles : LEACH, LEACH-SWDN et LEACH-C jusqu'à ce que le nombre de nœuds vivants dans le réseau soit supérieur à 70%. Cette évolution peut se justifier par le fait que ces différents protocoles gèrent un même nombre optimal de clusters car le seuil de sélection dépend de la probabilité p qui était définie à 5%. En effet, e-LEACH présente une bonne efficacité en terme du durée de vie, il est bien évident que le nombre de paquets délivrés augmente. A partir de ces résultats nous pouvons déduire qu'e-LEACH optimise efficacement l'équilibrage de l'énergie.

3.5.5 Conclusion

Nous avons proposé un protocole de routage hiérarchique basé sur le clustering permettant d'économiser et d'équilibrer l'énergie. L'objectif principal était d'améliorer le processus de sélection du "cluster-head" en considérant le niveau de réserve d'énergie et sa variance pour chaque nœud, ceci afin d'optimiser la durée de vie du réseau. Les résultats des simulations permettent de constater que le gain des performances d'e-LEACH augmente d'environ 25%. En revanche, à partir de 70% du nombre de nœuds survivants, la quantité de données reçues par la station de base diminue légèrement par rapport au protocole centralisé LEACH-C. Cette diminution

due à l'utilisation du paramètre "variance" car ce dernier ajuste progressivement le nombre de "cluster-heads" en fonction du nombre de noeuds survivants.

3.6 Le protocole décentralisé : LEACH-C2D

3.6.1 Introduction et Motivation

L'équilibrage énergétique (energy balance) dans l'ensemble des noeuds est une opération essentielle dans le RCSF pour prolonger la durée de vie du réseau. En effet, les algorithmes de clustering ont démontrés une meilleure efficacité en termes de réduction et de répartition de la consommation énergétique dans l'ensemble des noeuds. LEACH-C2D a été proposé pour l'optimisation de k-LEACH avec l'algorithme décentralisé. Ceci, dans le but de minimiser le surcout de communication et de calcul afin de trouver une efficacité et équilibre énergétique pour augmenter la durée de vie du réseau.

Dans les algorithmes centralisés tels que LEACH-C et k-LEACH, les cluster-heads sont sélectionnées par la station de base en utilisant les informations d'énergie restante et l'identification de tous les noeuds dans le réseau afin de déterminer l'énergie restante totale du réseau et ensuite d'identifier les noeuds cluster-head. Cette méthode nécessite donc des échanges périodiques des informations locales des noeuds, en l'occurrence leurs niveaux d'énergies et leurs identifications. Malgré tout, la méthode centralisée présente une meilleure performance en termes de durée de vie et gestion d'énergie que la méthode distribuée. Cela s'explique par le fait que le choix du "cluster-head" repose sur l'évolution de consommation d'énergie globale du réseau. Tandis que dans la méthode distribuée, l'auto-configuration s'avère nécessaire et devient l'une des caractéristiques indispensable lorsque le nombre de noeuds augmente. Cela nécessite parfois l'échange d'informations entre les noeuds. Ainsi, l'auto-configuration n'est pas nécessairement source d'économie d'énergie dans un réseau de capteur.

Pour profiter de l'avantage de l'algorithme centralisé k-LEACH et pour que le réseau soit plus *scalable*, on va élaborer un algorithme décentralisé de k-LEACH. Dans cette proposition, nous mettons l'accent sur la modification de l'élection du "cluster-head" pour que tous les noeuds puissent estimer l'énergie restante totale des noeuds dans le réseau dans le but d'obtenir un équilibre énergétique de l'ensemble des noeuds.

3.6.2 Sélection du "cluster-head" : Equilibrage d'énergie consommée

On rappelle que k-LEACH a été proposé pour améliorer le critère de choix du "cluster-head", tout en assurant qu'il existe k cluster dans le réseau. Notons que le "cluster-head" consomme plus d'énergie que les noeuds membres. Par conséquent, la détermination du nombre de clusters optimal permet de garantir l'efficacité d'énergie et d'assurer la couverture totale du réseau. C'est la raison pour laquelle, on a retenu

la fonction de seuil $p_i(r)$ pour l'élection "cluster-head" qui a été définie par :

$$p'_i(r) = k \frac{\gamma_i(r) * E_i(r)}{E_{tot}} \quad (3.40)$$

Où k : le nombre de clusters optimal, choisi de manière qu'il permet garantir la couverture de réseau. La fonction $\gamma_i(r)$ est un paramètre qui tient compte la dispersion de la consommation d'énergie. Elle est utilisée pour atteindre l'objectif principal qui est non seulement de maximiser la durée de vie du réseau de manière à minimiser la variance de l'énergie consommée de noeud mais aussi d'avoir une bonne distribution du "cluster-head". La considération de E_{tot} , comme si les "cluster-heads" sont élus parmi les noeuds qui minimisent la consommation globale d'énergie dans le réseau.

Evidemment, lorsque le réseau fonctionne, tous les noeuds n'ont pas la même quantité d'énergie résiduelle. En effet, LEACH-C2D prend en compte l'énergie résiduelle $E_i(r)$ des noeuds, par tour. D'une part, elle est nécessaire dans le but de choisir les noeuds qui ont plus de réserve d'énergie. D'autre part, elle est aussi nécessaire pour calculer le niveau de réserve d'énergie moyenne et la variance afin que le noeud puisse s'auto-proclamer comme cluster-head.

3.6.2.1 Détermination de la consommation d'énergie moyenne

Chaque noeud doit calculer leur niveau d'énergie, la moyenne et la variance de consommation d'énergie. Dans ce cas, tous les noeuds doivent stocker leur propre consommation, le niveau d'énergie restante actuelle et précédente pour déterminer la moyenne. Précédemment, dans e-LEACH, on a essayé d'équilibrer la consommation autour de la moyenne référence, en supposant que chaque noeud doit consommer la même quantité d'énergie moyenne à chaque tour. Pratiquement, cette hypothèse s'avère atteignable. En effet, il est préférable d'utiliser la méthode cumulative de la moyenne mobile pour calculer $\mu_i(r)$ dans le but de réduire le surcout de calcul et la mémorisation.

Si l'ensemble d'énergie résiduelle du noeud i par tour r est ordonné comme $\{E_i(1), E_i(2), \dots, E_i(r)\}$, et que la séquence de la moyenne mobile est $\{\mu_i(1), \mu_i(2), \dots, \mu_i(r)\}$. Alors, $\mu_i(r)$ peut être définie de manière récursive comme suit :

$$\mu_i(r) = \frac{(r-1) \cdot \mu_i(r-1) + E_i(r)}{r}, \quad (3.41)$$

Dans cette expression, tous les termes sont pondérés de façon égale dans sa valeur précédente. Cependant, nous pouvons également calculer des moyennes pondérées dans lesquelles les termes les plus récents sont donnés. Dans la pratique, il suffit de garder seulement la dernière moyenne de l'énergie résiduelle et son niveau de réserve d'énergie actuelle.

Enfin, nous pouvons calculer la variance selon le théorème de Koenig :

$$\mathcal{V}_i(r) = \frac{1}{R} \sum_{r=1}^R E_i^2(r) - \mu_i^2(r), \quad (3.42)$$

On pose :

$$\nu_i(r) = \frac{1}{R} \sum_{r=1}^R E_i^2(r),$$

On procède de la même façon que précédemment (3.41), on obtient :

$$\nu_i(r) = \frac{(r-1) \cdot \nu_i(r-1) + E_i^2(r)}{r}, \quad (3.43)$$

Par conséquent, la variance de l'énergie résiduelle peut être déterminée par :

$$\mathcal{V}_i(r) = \nu_i(r) - \mu_i^2(r), \quad (3.44)$$

Dans ces résultats d'analyse, on peut obtenir les valeurs exactes de la moyenne et la variance arithmétique.

3.6.2.2 Modélisation de la d'énergie restante totale du réseau

De (3.40), l'énergie totale qui reste dans le réseau est nécessaire pour l'expression du seuil. La méthode centralisée suppose que tous les noeuds transmettent à la station de base leur énergie résiduelle pour calculer la consommation moyenne ou totale d'énergie du réseau. Dans LEACH-C2D, la méthode décentralisée est l'une des solutions adoptées qui permet d'éviter le surcôt de communication supplémentaire entre les "cluster-heads" et la station de base. Dans ces conditions on peut se poser la question : comment un noeud connaît-il l'énergie restante de tous les autres noeuds dans le réseau ? *Broadcast schème* est une solution bien connue, si l'on suppose que tous les noeuds sont dans la portée de communication de l'autre [129]. Cependant, cela ne permet pas une efficacité énergétique dans le réseau de capteur. En effet, il est intéressant de fournir une méthode appropriée dans la perspective de l'équilibre énergétique.

Nous pouvons définir que, lorsque le réseau atteint la consommation d'énergie équilibrée, la durée de vie du réseau peut être définie par la ratio entre l'énergie initiale totale E_{IniTot} et la consommation énergétique globale E_{tot} du réseau :

$$T = \frac{E_{IniTot}}{E_{tot}} \quad (3.45)$$

En définissant que la consommation équilibrée d'énergie est atteinte dans le réseau lorsque les noeuds épuisent simultanément leur énergie, on peut en déduire que la

consommation globale du réseau à chaque tour doit être constante. En effet, l'énergie restante totale définie dans le réseau peut être exprimée de manière récursive par :

$$\sum_{i=1}^n E_i(r) = E_{IniTot} \left(1 - \frac{r}{T}\right) \quad (3.46)$$

L'énergie initiale totale dans un réseau composé de n noeuds ayant la même quantité E_o peut être déterminée par :

$$E_{IniTot} = n * E_o \quad (3.47)$$

Ainsi, l'expression de seuil (3.40) peut être réécrite comme :

$$p'_i(r) = k \frac{\gamma_i(r) * E_i(r)}{E_{IniTot} \left(1 - \frac{r}{T}\right)} \quad (3.48)$$

3.6.3 Limitation de l'équilibrage de la consommation d'énergie

D'après la relation (3.46), nous voyons clairement que l'évolution de la consommation globale du réseau décroît linéairement ainsi qu'en tenant compte les paramètres des simulations adoptées précédemment, et la relation (3.45), la durée de vie maximale du réseau peut atteindre est $T = 300$ tour (Fig.3.10).

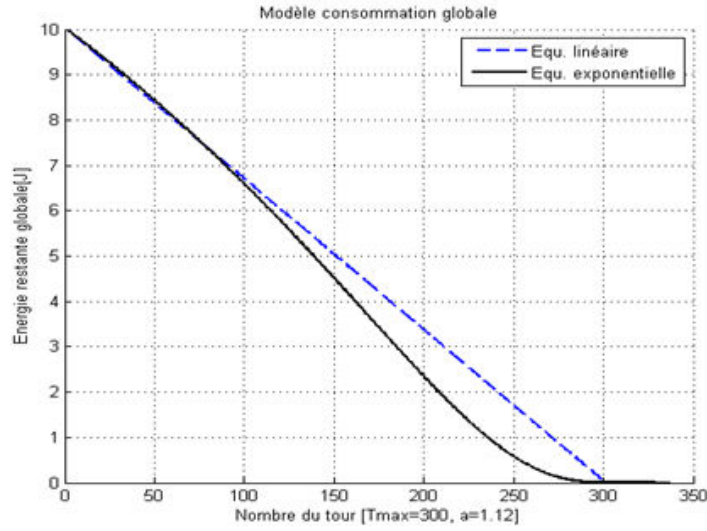
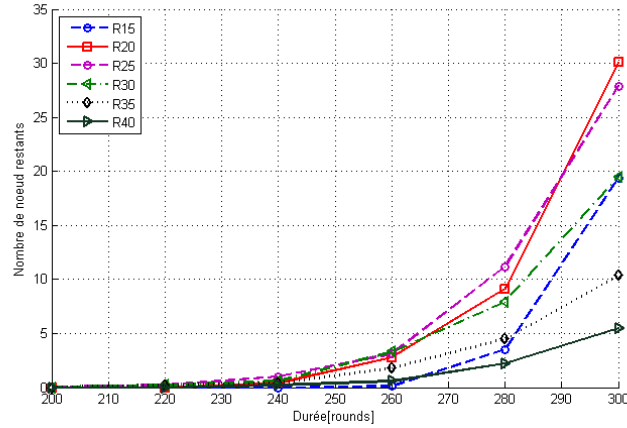
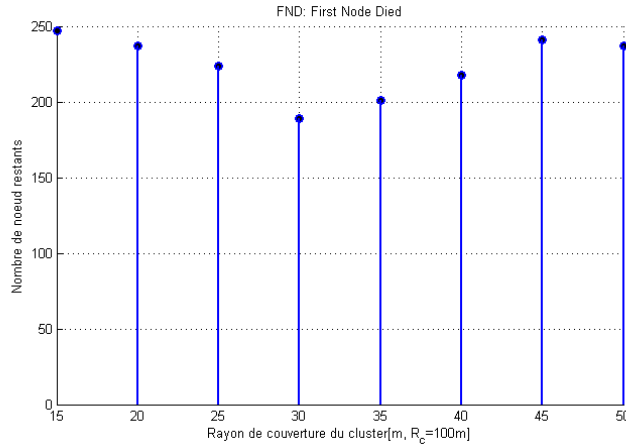


FIGURE 3.10 – Modèle de consommation

La figure 3.11.a nous montre le nombre de noeuds morts en respectant la variation de la portée de couverture du cluster. Nous remarquons qu'à 300 tour, le nombre de noeuds épuisés ne dépasse pas 30% de nombre de noeuds déployés dans le réseau, c'est-à-dire, il existe encore 70% noeud survivant. Cela vient du fait que la



(a) Nombre de Noeud épuisé de chaque rayon de couverture.



(b) Metrique FND : First Node Died.

FIGURE 3.11 – Durée de vie : modèle d'énergie globale fonction linéaire décroissante

probabilité de la sélection du "cluster-head" ne dépend seulement de l'énergie restante moyenne mais aussi le nombre moyen de clusters. En un mot, le nombre moyen de clusters est un paramètre du réseau à choisir judicieusement selon l'objectif.

La figure 3.11.b illustre la métrique FND représentant le temps du premier noeud qui meurt. Les résultats de cette figure nous montrent que pour le rayon de couverture des clusters appartenant à l'intervalle $[15, 25]$ et $[40, 50]$, le nombre de clusters minimal est compris dans l'intervalle $[5, 12]$ et $[1, 2]$ (voir. Fig.3.2), la durée de vie du noeud pour cette métrique atteint au minimum 200 tours. Exceptionnellement, pour le nombre de cluster $k = 12$, cette durée peut atteindre environ 249 tours.

L'énergie restante globale à 300 est illustrée dans la figure 3.12. A la lumière de cette figure, l'énergie restante globale du réseau est encore importante lorsque le rayon de couverture du cluster $R \geq 30$. Cela signifie que les noeuds consomment

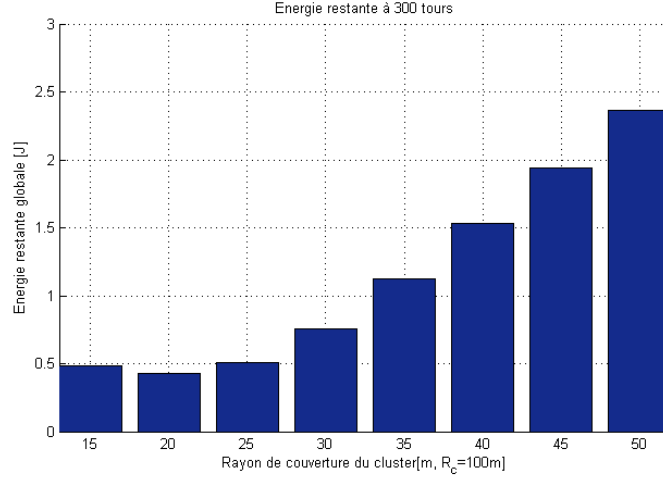


FIGURE 3.12 – Energie restante à 300 tours.

moins d'énergie lorsque le nombre de cluster diminue ($k \leq 5$). Comme démontré sur cette figure, cette économie d'énergie augmente à plus de 50% par rapport au cas où $R < 30$ ou $k > 5$. En effet, la technique d'équilibrage de la consommation énergétique du réseau, dans l'objectif que tous les noeuds épuisent simultanément leur énergie, ne peut être atteinte qu'en augmentant le nombre de cluster dans le réseau. Toutefois, en augmentant ce nombre de cluster, la durée du temps allouée à un noeud pour accéder aux ressources va augmenter, ce qui implique par la suite l'incrémentation du temps d'attente des "cluster-heads". En outre, si nous considérons le nombre de cluster optimal $R \geq 30$, le réseau peut tourner au delà de 300 pour ne pas gaspiller l'énergie.

En guise de conclusion, compte tenu de ces résultats, l'équilibrage de la consommation énergétique (dans le sens que tous les noeuds épuisent simultanément leur énergie) dans le réseau, basée sur la technique de clustering héritante une communication "one-hop", n'est n'améliore pas l'efficacité énergétique dans le réseau. Pour éviter ce gaspillage énergétique, nous avons mis en place un modèle de consommation qui décroît exponentiellement en fonction du temps, et basé sur le principe du modèle linéaire (Fig.3.10).

3.6.4 Modèle énergétique global exponentiel

On a défini une fonction qui permet de décrire l'évolution de la consommation d'énergie globale du réseau qui décroît exponentiellement au cours du temps (round)(Fig.3.10). Cela peut être définie par :

$$\sum_{i=1}^n E_i(r) = E_{IniTot} \times e^{-\left(\frac{r}{aT-r}\right)} \quad (3.49)$$

Avec, a est une constante de régression, qui dépend les paramètres de simulation et la métrique FND pour le protocole à décentraliser.

T : La durée de vie définit par la relation (3.45).

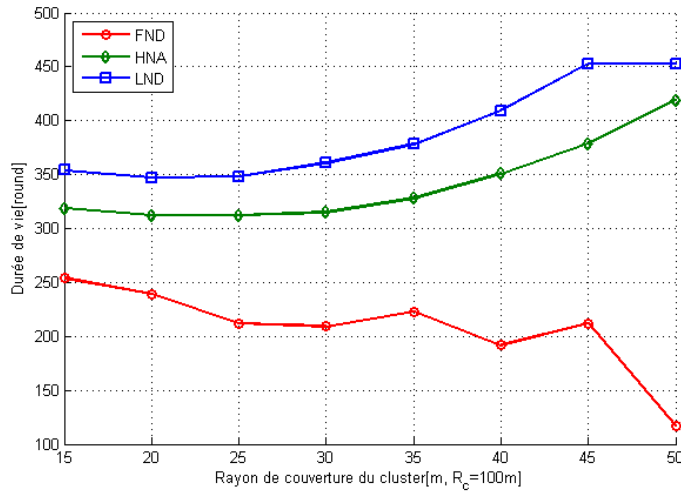
3.6.5 Résultats : Modèle de consommation globale exponentielle

Pour les simulations, nous avons retenu les différents paramètres et les configurations du réseau créés précédemment lors de la simulation du k-LEACH et k-LEACH-VAR. Dans un premier temps, nous avons fait varier le rayon de cluster afin d'évaluer les performances de LEACH-C2D. En suite, nous avons considéré le rayon de couverture du cluster $R = 35$ en vu de comparer avec l'algorithme centralisé k-LEACH, LEACH, LEACH-C et LEACH-SWDN.

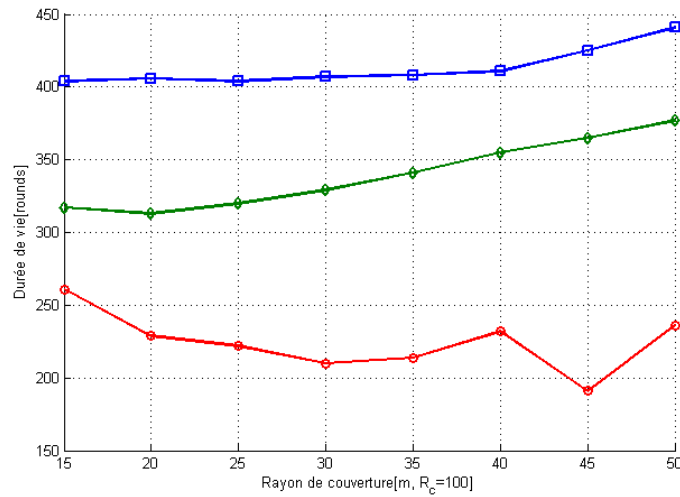
Les résultats de la figure 3.13 représentent la comparaison de la durée de vie du réseau en fonction de la variation du rayon de couverture du cluster pour k-LEACH et k-LEACH. Nous constatons globalement que K-LEACH-C2D a de meilleures performances en termes de durée de vie du réseau pour les trois métriques FND, HNA et LND que k-LEACH. Cela vient du fait que dans K-LEACH, tous les noeuds transmettent périodiquement leur position et leur identification à chaque début de tours. Cela a pour conséquence une surconsommation d'énergie du réseau entier. Néanmoins, l'utilisation de modèle de consommation globale du réseau dans K-LEACH-C2D lui permet de surpasser la durée de vie du k-LEACH.

La figure 3.14 présente le nombre de noeuds restants en fonction du temps (tours), pour les cinq différents algorithmes de routage : LEACH, LEACH-C, LEACH-SWDN, k-LEACH et K-LEACH-C2D. Les résultats de cette figure nous démontrent que K-LEACH-C2D a de meilleures performances que les autres protocoles. La performance de K-LEACH-C2D par rapport K-LEACH peut s'expliquer par le fait que, K-LEACH-C2D a pu éliminer le coût de la communication privilégiée par l'envoi périodique des informations locales de chaque noeud à la station de base au cours de chaque phase de l'installation, qui n'est pas négligeable pour k-LEACH. En considérant, par exemple, la métrique HNA permet d'atteindre 348 tours, alors que LEACH, LEACH-C, LEACH-SWDN et k-LEACH atteignent respectivement 258, 300, 316 et 330 tours. La durée de vie est améliorée de 34% et de 16% pour K-LEACH-C2D par rapport aux deux protocoles distribués LEACH et LEACH-SWDN, respectivement.

L'amélioration apportée par K-LEACH-C2D est encore confirmée par les résultats de la figure 3.15. Cette figure illustre l'énergie restante globale des noeuds pour les cinq algorithmes en fonction du temps (tours). A part, l'explication que nous avons apportée précédemment, l'utilisation du paramètre variance permet d'avoir la bonne distribution du cluster-head, donc la distance de communication intra-cluster diminuant, cela permet de diminuer la consommation d'énergie des noeuds



(a) K-LEACH.



(b) K-LEACH-C2D.

FIGURE 3.13 – Illustration de la durée de vie de k-LEACH et K-LEACH-C2D en fonction de rayon de couverture

du réseau.

La figure 3.16 montre le nombre de paquets de données reçues par la station de base en fonction du nombre noeud survivant du réseau. Les résultats dans cette figure illustrent l'efficacité d'K-LEACH-C2D en livrant un nombre plus élevé de message de donnée que des protocoles : LEACH et K-LEACH. Cette évolution peut s'expliquer par le fait que k-LEACH-C2D génèrent un nombre de cluster quasi-stable vue que le seuil de la sélection du cluster-head dépend aussi la consommation globale du réseau, cette consommation était modélisée par la fonction décroissante

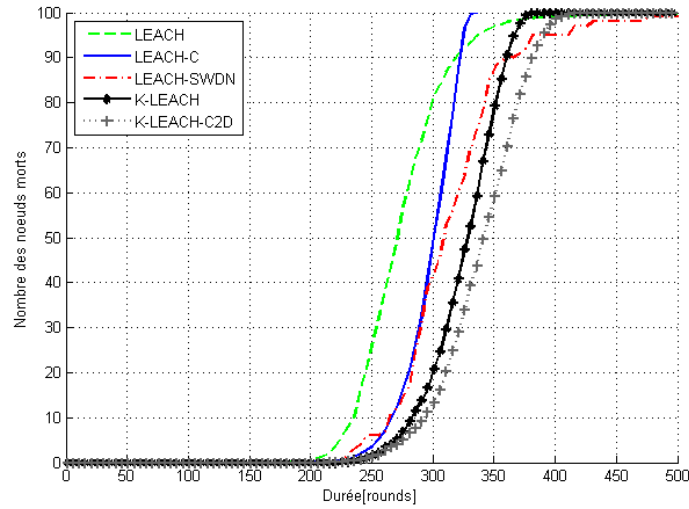


FIGURE 3.14 – Comparaison de la durée de vie de k-LEACH-C2D

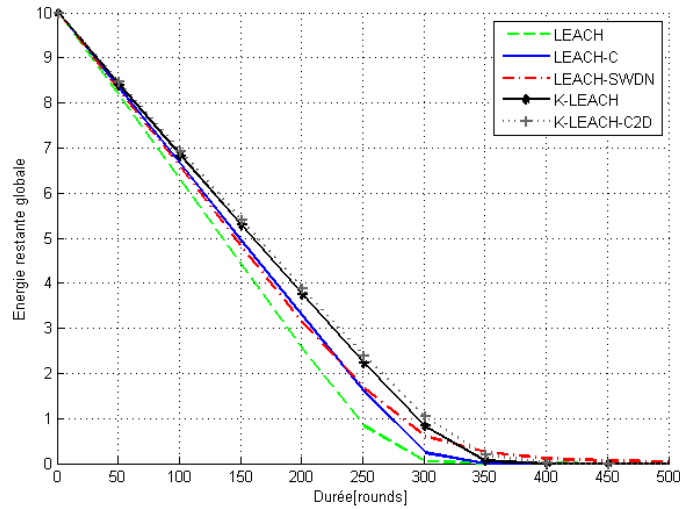


FIGURE 3.15 – Comparaison d'énergie restante de k-LEACH-C2D

exponentielle au cours du temps, évidemment, cette fonction évolue de manière stable. De plus, K-LEACH-C2D présente une bonne efficacité en terme du durée de vie par rapport à ces deux protocoles, il est bien évident que le nombre de paquet délivrés augmente. En revanche, le protocole centralisé LEACH-C, présente une quantité supérieure de donnée délivrée à la station de base, car LEACH-C génère plus nombre de cluster que K-LEACH-C2D.

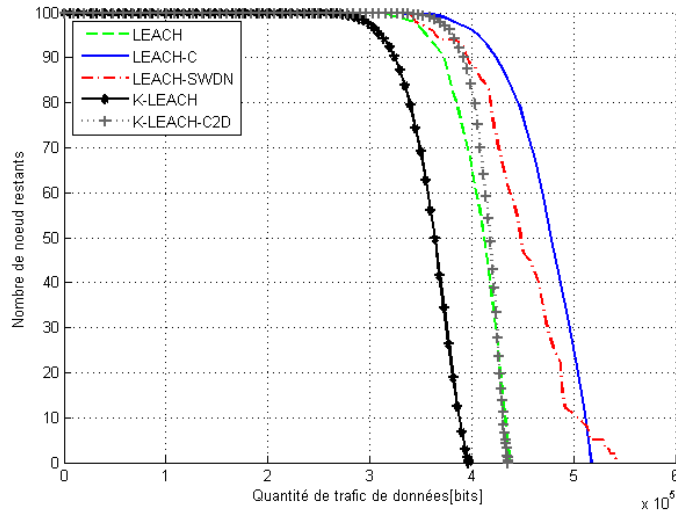


FIGURE 3.16 – Comparaison des traffics au sein du réseau k-LEACH-C2D

3.6.6 Conclusion

Dans cette section, nous avons proposé deux versions d'un protocole k-LEACH avec une méthode décentralisée, notée K-LEACH-C2D, afin d'améliorer l'efficacité énergétique ainsi que l'équilibre énergétique dans l'ensemble du réseau. La première proposition repose sur l'objectif d'équilibrage de la consommation énergétique de tous les nœuds, en ciblant le fait que tous les nœuds doivent être morts simultanément. Les résultats nous montrent que l'équilibre énergétique nécessite une augmentation du nombre de clusters. En revanche, quand le nombre de cluster diminue, le système conserve d'énergie.

La deuxième proposition consiste donc à proposer un modèle de consommation globale qui décroît exponentiellement au cours du temps, pour ne pas gaspiller cette énergie. Cette proposition permet d'ajuster l'évolution de la consommation globale suivant l'énergie initiale totale du réseau. Grâce à ces deux modèles, nous avons pu supprimer l'échange périodique des informations locales des nœuds avec la station de base, le protocole K-LEACH-C2D réduit le trafic de contrôle au sein du réseau permettant de conserver l'énergie.

Les résultats montrent que K-LEACH-C2D offre de meilleures performances en termes de la durée de vie du réseau, la consommation d'énergie ainsi que le nombre de paquets délivrés à la station de base par rapport aux autres protocoles distribués.

3.7 Cluster-Based Multihop Protocols

3.7.1 Motivation

En général, un réseau de capteur sans fil se compose d'un grand nombre de petits et moyens noeuds et d'une station de base. Habituellement, les noeuds contrôlent leur environnement, collectent les données captées et les envoient à la Station de Base. Cependant, ils sont limités en termes source d'énergie, de puissance de traitement et de capacité de stockage mémoire. Ces contraintes doivent être bien gérées dans la conception de protocole communication. Typiquement, un noeud de capteur est déployé dans une zone hostile où il est difficile de remplacer les batteries (intégrées) épuisées. Par conséquent, l'énergie devient alors la source fondamentale de contraintes devant être contrôlées attentivement dans le but de prolonger la durée de vie du réseau de capteur.

Les algorithmes de routage précédemment conçus basés sur les clusters ont affirmé que la technique de clustering est une technique importante qui peut organiser le réseau en hiérarchie connectée, l'équilibrage de charge et augmenter la durée de vie du réseau dans un réseau de capteurs sans fil. La plupart des algorithmes de clustering proposés se basent sur l'élection de cluster-head parmi les noeuds au sein du réseau. Le fait de choisir les cluster-heads parmi les noeuds ayant une réserve d'énergie élevée et faible variance, et de faire tourner le rôle du cluster-head de manière périodique étend considérablement la durée de vie du réseau. Cependant, le problème de la consommation déséquilibrée d'énergie perdure. Cela s'explique qu'au moins deux raisons : le déploiement aléatoire des noeuds et le contrôle de puissance de transmission qui est proportionnelle à la portée de transmission. Or le cluster-head est non seulement responsable de la transmission directe des données agrégées à la station de base mais aussi de la collecte des données captées provenant de des membres du cluster. Par conséquent, les noeuds les plus éloignés de la station de base meurent rapidement dû à la longue distance de transmission. Effet, en réduisant la distance de transmission des noeuds entre la station de base, la dissipation d'énergie sera minimisée.

C'est pourquoi, des nombreux travaux focalisent aujourd'hui leur attention sur la gestion de l'énergie consommée par les noeuds en limitant la portée de transmission autant que possible à un certain seuil. En effet, les tendances de conceptions des protocoles, à l'heure actuelle, dans les réseaux sans fils s'appuient sur la technique de routage multi-sauts [52, 53, 130], clustering [56, 100, 131], et en faisant la combinaison de ce deux protocoles d'où l'épanouissement des protocoles hybrides [36, 68, 86].

Les protocoles de routage multi-saut sont conçus au moins pour deux objectifs : Permettre l'extension du réseau des capteurs sans fil en présence d'un grand nombre de noeuds et trouver un chemin optimal pour améliorer l'efficacité d'énergie pour une transmission fiable de données vers la station de base en passant par des noeuds

intermédiaires. Cependant, la communication multi-saut tend à emprunter le même chemin vers la station de base, cela conduit rapidement à un épuisement prématuré des noeuds constituant la route "préférée", en l'occurrence les noeuds plus proches de la station de base. Ceci conduit à l'existence de trous d'énergie aux alentours de la station de base [59, 132]. Dans les deux modes de communication : multi-saut et clustering qui hérite la communication directe (one-hop), le problème de consommation déséquilibrée d'énergie parmi les différents noeuds reste inévitable, conduisant à la situation où certains noeuds qui utilisent leur énergie à des taux plus élevés, qui vont épuiser leur énergie plus rapidement que d'autres. En effet, une méthode de routage avec une consommation d'énergie équilibrée devrait être trouvée.

Comme mentionné dans [131], les transmissions courtes (avec un petit nombre de sauts) sont généralement plus efficace en énergie que la transmission directe. Durant la dernière décennie, des travaux font état de méthodes couplant les algorithmes de clustering avec des routages multi-sauts intra ou inter-cluster dans le but de minimiser la distance de transmission. Dans cette section, nous proposons deux protocoles hybrides pour les réseaux de capteurs sans fil en réduisant les problèmes considérés.

La suite de section est structurée comme suit : la sous-section 3.7.2 décrit le protocole FRSM (Forwarding and Routing Stateless Multihop Protocol), qui a pour objectif d'évaluer la méthode de couplage entre le clustering, le routage multi-saut ainsi que la méthode d'agrégation en vue de minimiser la consommation énergétique du cluster-head afin d'augmenter la durée de vie du réseau. En suite, dans la sous-section 3.7.3 décrit le protocole CB-RSM (Cluster-Based Routing Stateless Multihop Protocol). Une méthode d'ordonnancement est proposée entre les clusters et la station de base afin que les cluster-heads s'auto-organisent pendant la phase de transmission des données (inter-cluster) en vue de rendre le réseau plus *scalable*.

3.7.2 Forwarding and Routing Stateless Multi-Hop Protocol

3.7.2.1 Introduction

Cette section présente l'évaluation d'un nouveau protocole de routage hiérarchique dans les RCSFs, dénoté FRSM "Forwarding and Routing Stateless Multihop protocole" [91]. Comme on a évoqué précédemment, la réduction de consommation d'énergie peut être améliorée au moyen de protocoles de routage de clustering dans un réseau de capteur sans fil. En effet, FRSM fait l'objet d'une étude d'un protocole de routage multi-saut entre clusters (intra-cluster) afin d'optimiser efficacement la dépense énergétique du noeud lors de la communication, FRSM s'appuie sur les mécanismes suivants : (1) la formation des clusters se fait périodiquement sous contrôle du "cluster-head", (2) la communication entre cluster-head et la station de

base se fait en multi-saut si cette dernière ne peut pas atteindre directement, (3) dans l'optique d'amélioration de l'efficacité, tout se fait en considérant la technique d'agrégation progressive au cours de route et une requête séquentielle des données.

3.7.2.2 Idée de base et concept du FRSM

L'algorithme de FRSM est proposé dans le but d'augmenter la durée de vie et de balancer la consommation d'énergie du réseau en minimisant la distance de communication des noeuds. L'idée principale de FRSM est d'injecter le protocole de routage géographique dans le protocole à base clustering. Le routage multihop intervient entre les cluster-heads (communication inter-cluster) vers la station de base pour remédier au problème de longue transmission. Pour éviter l'emprunt systématique des chemins passant par les noeuds se trouvant au plus proche de la station de base, on considère la rotation périodique de rôle du cluster-head. En effet, le cluster-head le plus loin de la station de base doit réagir comme la première source d'information.

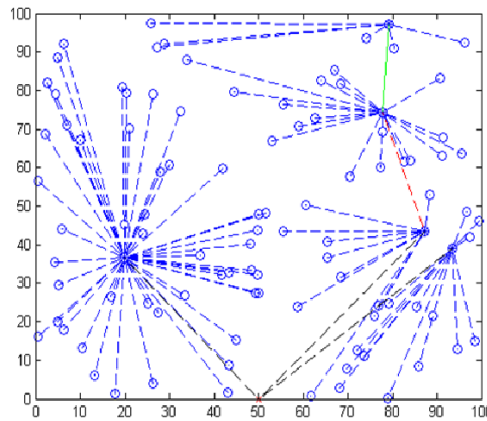


FIGURE 3.17 – Architecture du réseau dans FRSM

D'une part, le protocole GPSR ("Greedy Perimeter Stateless Routing") [52, 130] ayant pour objectif d'élever l'extensibilité du réseau en présence d'un grand nombre de noeuds. Le principal avantage réside dans le fait qu'il nécessite seulement que les positions des leurs voisins directs pour garantir l'acheminement des données, sans garder les différents états au cours de routage (Stateless). En effet, cela minimise l'utilisation des ressources mémoire et le surcout de calcul en nombre d'itération nécessaire pour parcourir la table de routage. GPSR fait appel souvent à l'algorithme "Greedy Forwarding" qui consiste à élire un noeud voisin plus proche de la destination. De ce fait, les noeuds aux alentours de la station de base souffrent en supportant beaucoup plus de trafic, cela génère un risque de mort prématurée des noeuds en l'occurrence les noeuds plus proches de la station de base. D'autre

part, LEACH hérite du routage à transmission directe entre le "cluster-head" et la station de base. C'est pourquoi, FRSM propose la combinaison de ces deux méthodes en désignant le "cluster-head" le plus loin de la station de base comme la première source d'information.

3.7.2.3 Algorithme de FRSM

Notre algorithme suppose que la station de base connaît la position de tous les noeuds déployés dans le réseau. De plus, les noeuds rendent compte de leur voisinage direct ("one-hop") après formation du cluster et sont capables de mesurer leurs distances par rapport à leurs voisins. Il est bien noté que des mesures très précises de calcul de distance sont possibles, grâce à l'utilisation des "rssi" et "lqi" [133], ultrasons [], Médusa [], ...etc.

En effet, le mécanisme de transmission de données dans FRSM peut se décomposer en deux phases, la communication *intra-cluster* et la communication *inter-cluster*. L'opération dans la communication intra-cluster est similaire à LEACH, après que la base station ait sélectionné les "cluster-heads". Tandis que dans la communication inter-cluster, la station de base interroge le "cluster-head" plus loin d'elle, ainsi ce "cluster-head" devient première source d'information (S).

A partir de ce moment, l'algorithme GPSR intervient en utilisant le mode "Greedy Forwarding", la source (S) choisit et envoie les données agrégées au meilleur voisin (qui est le "cluster-head" le plus proche de la destination). Ensuite, le "cluster-head" destination devient la source d'information, il cherche le prochain meilleur voisin. Ce cycle se poursuit jusqu'à ce qu'il échoue dans le cas où il n'y a pas de voisins. A partir de ce moment, le noeud ajuste sa puissance de transmission et l'algorithme "Permit Forwarding" intervient, qui consiste à envoyer les données au "cluster-head" qui est plus proche de la source. Et cela continue jusqu'à ce que les données arrivent à la station de base.

Le cluster-head qui n'est pas encore interrogé ou qui ne participe pas pendant l'acheminement, le BS interroge à nouveau. Cependant, tous les cluster-heads après avoir transmis les données éteignent leur radio. Par conséquent, il est fort probable que les cluster-heads ne puissent pas trouver un voisin. Dans ce cas, les données seront envoyées directement à la station de base. Cette hypothèse est acceptable dans le fait qu'on envisage à comparer notre algorithme à LEACH en vue d'évaluer l'importance de cette méthode.

3.7.3 Distributed Cluster-Based Routing Protocol

3.7.3.1 Introduction et Motivation

Au vu des problèmes de passage à l'échelle du réseau, l'auto-configuration et la communication multi-saut au niveau du réseau semblent incontournables. En effet,

L'algorithme FRSM combine la rotation périodique de cluster-head et la communication multi-saut. En considérant que le cluster-head ne retransmet qu'une seule fois au cours d'un cycle, ce principe permet non seulement d'équilibrer la consommation d'énergie entre les noeuds mais aussi de résoudre l'existence de trous d'énergie autour de la station de base, car cela évite d'acheminer un paquet pour un même chemin préféré vers la station de base. Toutefois, FRSM ne prend pas en compte l'énergie restante du noeud. Par ailleurs, il suppose que la base station soit accessible en un seul-saut dans le cas où les "cluster-heads" ne trouvent pas un voisin dans sa portée prédéfinie. De plus, la méthode de requête séquentielle de la base station peut engendrer une latence.

Dans cette sous section, nous proposons un protocole distribué dénoté "Cluster-Based Routing Stateless Multihop" (CB-RSM) pour les réseaux de capteurs sans fil, permettant d'atténuer les problèmes mentionnés. Le CB-RSM est un protocole efficace en énergie qui combine la rotation de sélection de cluster-head et une méthode de routage multi-saut de données, afin de prolonger la durée de vie du réseau. La sélection cluster-head dépend de l'énergie résiduelle et de la variance d'un noeud. Dans la transmission de données entre les clusters et la station de base, une méthode d'ordonnancement, nommé "Binary Greedy Forwarding" est proposée. Cela consiste à établir une communication entre les clusters et la station de base afin que les cluster-heads s'auto-organisent pendant la phase de transmission des données en vue de rendre le réseau plus scalable. L'acheminement des données est basé sur la construction du chemin "élémentaire" qui ne passe qu'une fois par un noeud. Pour se faire, on associe une notion de poids à chaque noeud, celle-ci est en fonction de l'énergie restante et de la distance entre le "cluster-head" et la station de base.

3.7.3.2 Sélection des "Cluster-heads"

La position du "cluster-head", telle que la distance séparant le "cluster-head" et la station de base au sein du réseau, a un impact assez important sur la variation de la consommation totale d'énergie. Sélectionner les "cluster-heads" parmi les noeuds ayant une énergie restante élevée ainsi que de faire tourner le rôle du "cluster-head" périodiquement entre les noeuds étendent considérablement la durée de vie du réseau [6]. Cependant, le problème de la consommation déséquilibrée d'énergie perdure à cause du déploiement aléatoire des noeuds. C'est pourquoi, e-LEACH a tiré les "cluster-heads" parmi les noeuds possédant une valeur moyenne d'énergie restante élevée et une petite variance afin de réduire la consommation déséquilibrée d'énergie, et rendre à bien distribuer les "cluster-heads" dans le réseau [58].

C'est pourquoi, la méthode de sélection des "cluster-heads" d'e-LEACH est retenue ainsi que l'adoption d'un routage multi-saut entre les "cluster-heads" afin de diminuer la consommation énergétique du noeud.

3.7.3.3 La communication intra-cluster

Le mécanisme de communication de données au sein du réseau entier inclut la transmission de données "inter-cluster" et "intra-cluster" (Fig.3.18).

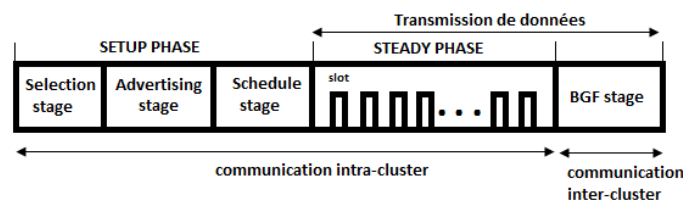


FIGURE 3.18 – Opération dans le protocole CB-RSM

L'organisation de la *communication intra-cluster* est similaire au protocole LEACH. Une fois que le "cluster-head" a été sélectionné, le protocole CSMA est utilisé pour diffuser les messages d'annonce "**AdvMsg**" à tous les noeuds du réseau. Et puis, tous les noeuds estiment leurs distances entre les cluster-heads émetteurs basé sur l'indicateur de puissance du signal reçu (RSSI). Ensuite, les noeuds envoient un message "**Join-Request**" au "cluster-head" le plus proche en utilisant encore le protocole CSMA. Après que le cluster head ait reçu tous les messages d'information, il utilise TDMA pour planifier la réception de données qui est basé sur le nombre de noeuds membres dans le cluster, et envoyer par la suite la notification à ces dernières.

3.7.3.4 Aperçu de la communication inter-cluster dans CB-RSM

La phase de la "**communication inter-cluster**" consiste à l'acheminement de données entre les "cluster-heads" en occurrence les plus éloignés de la station de base. Actuellement, il existe plusieurs possibilités, telle que la méthode "**spanning tree**" [134]. Mais cette méthode tend à surcharger (overload) les noeuds plus proches de la destination (Fig.3.19.A); tandis que FRSM propose un algorithme itératif pour construire un chemin unique avec une seule source vers la station de base (Fig.3.19.B). Cependant, FRSM a proposé que les cluster-heads qui sont déjà participé à la retransmission de données doivent éteindre son module radio, ceci peut générer un risque d'existence d'un "cluster-head" isolé. Par conséquent, si la station de base n'est pas accessible à un seul saut, les données seront perdues. De ce fait, CB-RSM présente l'algorithme "BGF", qui tend à remédier ce problème. En vue d'équilibrer la consommation d'énergie parmi les cluster-heads. Ces dernières envoient les données vers la station de base via technique multi-saut ou direct. (Fig.3.19.C).

Dans la perspective d'efficacité énergétique, l'algorithme BGF consiste en la conception d'un algorithme d'ordonnancement entre les clusters et la station de base afin que les "cluster-heads" s'auto-organisent pendant la phase de transmission de

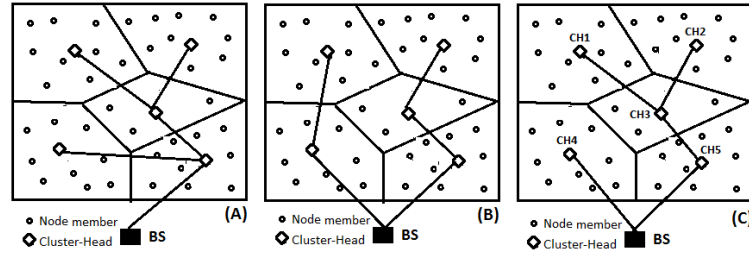


FIGURE 3.19 – Architecture du réseau : (A) spanning tree, (B)FRSM et (C) CB-RSM

données "inter-cluster" en vue de rendre le réseau plus scalable. En effet, le "cluster-head" ne garde seulement dans leur table de routage que l'ID de leur successeur et le nombre total de leur prédécesseur. Cette dernière est utile par le fait le "cluster-head" estime s'il doit participer ou pas dans la retransmission de données. En effet, si le "cluster-head" n'a pas de prédécesseur, il peut décider de manière autonome à se communiquer à leur successeur.

3.7.3.5 L'algorithme "Binary Greedy Forwarding"

L'algorithme "Binary Greedy Forwarding" exécute dans la phase de transmission de donnée "intra-cluster" (Fig.3.18). En effet, pour qu'on puisse élaborer cette algorithme, considérons d'abord le graphe de la figure.3.19.C. On pourra associer une tableau à ce graphe (table.3.3), appelé matrice booléenne, qui indique l'existence d'un lien entre deux "cluster-heads". Le principal objectif n'est autre que de construire un chemin élémentaire qui ne passe qu'une seule fois par un "cluster-head" (Fig.3.19.B). Cependant, il existe un cas critique, par exemple, le CH_4 n'est pas accessible directement par CH_1 . En fait, CH_1 doivent passer par CH_3 . De plus, le noeud CH_3 doit passer l'un des noeuds CH_4 et CH_5 . Pour faciliter l'analyse, on adopte la table binaire (m_{ij}) qui peut définir de la manière suivante : on met 1, si le lien (CH_i, CH_j) existe. Dans le cas contraire on écrira 0.

A la lumière du tableau 3.3, la colonne représente la table des prédécesseurs et la ligne représente la table des successeurs. D'abord, en raisonnant sur la table de prédécesseurs, si le lien n'existe pas (*que des 0*), cela veut dire que le noeud ne possède pas de prédécesseur, donc les noeuds peuvent réagir comme source (CH_1 et CH_2). Si on suppose que CH_5 est choisi par le CH_3 comme successeur, alors CH_4 peut réagir aussi comme source. Par contre, pour les CH_3 et CH_5 , leur table de prédécesseur est marquée par 1, donc ils doivent attendre leurs prédécesseurs correspondants. Ensuite, si on regarde maintenant la table de successeur (ligne), on voit clairement que si le lien n'existe pas (BS exclus), les noeuds peuvent réagir

	CH_1	CH_2	CH_3	CH_4	CH_5	BS
CH_1	0	0	1	0	0	0
CH_2	0	0	1	0	0	0
CH_3	0	0	0	0/1	1	0
CH_4	0	0	0	0	0	1
CH_5	0	0	0	0	0	1
BS	0	0	0	0	0	0

TABLE 3.3: Matrice booléenne d'existence du lien

comme puits (CH_4 et CH_5), ils doivent attendre leurs prédécesseurs correspondants. Dans le cas contraire, ils possèdent successeur. L'algorithme suivant donne l'aperçu global de l'établissement d'un lien entre les cluster-heads.

Algorithm 1: L'algorithme de BGF

Input: $NP[i]$: Nombre de prédécesseur du CH_i et Ω : l'ensemble de CH

Result: Construction ordonnée du lien entre le CHs

```

1 for  $i \leftarrow 1$  to  $Card(\Omega)$  do
2   if  $NP[i] = 0$  then
3     for  $j \leftarrow i + 1$  to  $Card(\Omega)$  do
4       if  $m_{ij} = 1$  then
5         Etablir le lien entre  $(CH_i, CH_j)$ 
6          $NP[j] = NP[j] - 1$ 
7       end
8     end
9   end
10 end
```

Dans le cas pratique, le noeud ne mémorise qu'une table qui contient l' Id de leur successeur (SID) et le nombre de son prédécesseur (NP). Cette méthode porte d'avantage sur la contrainte de mémoire et le nombre d'itérations de calcul. Par exemple, le CH_3 ne garde que le nombre de ses prédécesseurs et l' Id de son successeur, soit respectivement $NP = 2$ et $SID = 5$. Ce noeud doit attendre les données via CH_1 et CH_2 . Le décomptage de prédécesseur se fait lorsqu'il reçoit les données via ses voisins. Dès que le nombre de prédécesseur NP vaut 0, il peut transmettre ses données vers CH_5 .

3.7.3.6 Formulation et Construction du table "Binary Greedy Forwarding"

Le problème est de comment former une matrice binaire qui, non seulement, permettrait d'ordonner les "cluster-heads" mais également permettrait d'identifier

facilement leur voisin direct ? Une notion du poids est introduite en vue d'ordonner les noeuds "cluster-heads" afin que les noeuds plus éloignés de la station de base puisse réagir comme une première source sans intervention de la station de base. De ce fait, chaque noeud évalue son propre poids, qui peut être calculée comme suit :

$$\psi_i(r) = (1 - \frac{R}{D_i(rssi)}) \frac{E_i(r)}{E_o} \quad (3.50)$$

Avec $E_i(r)$ désigne la valeur de l'énergie restante du cluster-head i , $D_i(rssi)$ estimation de la distance à la station de base selon la valeur du RSSI et R la portée radio.

Dans la relation (3.50), l'énergie restante du noeud est biaisée par leur distance par rapport à la station de base. Par conséquent, la valeur du poids dépend fortement de la distance. La nécessité de l'énergie restante est incontournable lorsque le noeud devrait choisir une route parmi leur voisin, dans le cas où, la distance par rapport à la station de base de ces dernières est égale. D'après cette relation, deux cas pourraient être envisagés :

Cas 1. Si la distance $D_i(rssi)$ est inférieure ou égale au rayon de communication du cluster R , alors la valeur de $\psi_i(r)$ devient négatif ou nulle. Donc, la station de base est dans le rayon de couverture du cluster. En effet, le noeud peut communiquer directement à la station de base à moins qu'il n'y a pas de prédécesseur.

Cas 2. Si la distance $D_i(rssi)$ est strictement supérieure au rayon de communication du cluster R , alors la valeur de $\psi_i(r)$ est surement positif. De plus, si le noeud est loin de la station de base, la valeur du poids augmente. Ainsi, le noeud doit communiquer à multi-saut.

En effet, cette expression permet au cluster-head de choisir l'un de ses voisins qui possède un poids élevé. Ce choix consiste à construire un chemin élémentaire et pour éviter également la surcharge du cluster-head plus proche de la station de base. Par conséquent, si on définit que Ω_i étant l'ensemble de "cluster-head" voisins du CH_i et Ψ_i l'ensemble de ces poids, on peut définir la matrice binaire comme suit :

$$m_{ij} = \begin{cases} 1, & \text{si } \psi_j = \max_{k \in \Psi_i}(\Psi_k) \\ 0, & \text{sinon} \end{cases} \quad (3.51)$$

En d'autres termes,

$$route(CH_i, CH_j) = \{(CH_j, \psi_j) \in \Omega_i \times \Psi_i, i \neq j \mid \psi_j = \max(\Omega_i)\} \quad (3.52)$$

Ces résultats d'analyse nous permettent de conclure que l'expression (3.50) peut ordonner l'ordre de position de cluster-head. En effet, le "cluster-head" plus loin de la station de base peut réagir comme première source d'information.

3.7.3.7 Principe du protocole basé sur l'algorithme BGF

Etant donné qu'une fois la phase de sélection du cluster-head soit terminé, les "cluster-heads" diffusent un message d'annonce "AdvMsg" à tous les noeuds du réseau afin que ces derniers joignent le "cluster-head" plus proche d'eux. Les "cluster-heads" se mettent souvent en écoute passive (Idle mode) pour pouvoir accéder au medium de transmission (la technique CSMA) ce qui génère une consommation importante. En fait, afin de ne pas gaspiller cette énergie ainsi que d'éviter une autre découverte de route après la formation du cluster (comme dans [94, 134, 135]), le message d'annonce "AdvMsg" doit contenir l'*Id* et le poids du noeud, afin que les autres "cluster-heads" déterminent leur prédécesseur (*NP*) et l'ID de leur successeur immédiat (*SID*). Compte tenu des contraintes de mémoire et de la capacité de calcul, les "cluster-heads" ne mémorisent que l'*Id* de son successeur et le nombre de son prédécesseur. Alors, un principe de comptage de prédécesseur se fait localement à chaque fois que le noeud entend le message "AdvMsg". L'algorithme suivant porte une vue globale de détermination de l'*Id* de leur successeur et le nombre de leur prédécesseur *NP*, cette algorithme s'exécute durant la phase de message d'annonce "Advertinsg stage" du setup-phase de la communication intra-cluster (Fig. 3.18).

Algorithm 2: Détermination du SID et NP dans BGF

Input: $\psi_i(r)$ du CH_i
Result: $SID[i]$:ID de successeur, $NP[i]$:Nombre de prédécesseur du CH_i

```

1  max  $\leftarrow$  0
2   $NP[i] \leftarrow$  0
3  Broadcast AdvMSG(ID,  $\psi_i$ )
4  if  $\psi_i \leq 0$  then
5    |  $SID[i] \leftarrow$  BS.Id
6  else
7    | while ("Advertising stage" n'est pas terminé) do
8    |   Recevoir AdvMSG(ID,  $\psi_i$ )
9    |   if  $\psi_i > \psi_j$  then
10   |   | if max  $\leq \psi_j$  then
11   |   |   | max  $\leftarrow \psi_j$ 
12   |   |   |  $SID[i] \leftarrow CH_j.Id$ 
13   |   | end
14   |   else
15   |   |    $NP[i] = NP[i] + 1$ 
16   |   | end
17   | end
18 end
```

Notons que si un "cluster-head" reçoit des données via à un autre "cluster-

head", il retransmettra les données agrégées et les données reçues directement vers son prochain saut sans fusionner les données.

3.7.4 Simulations et Discussions

Pour les simulations, nous avons retenu les configurations du réseau créé précédemment lors de la simulation du k-LEACH et k-LEACH-VAR. Par ailleurs, le rayon de couverture du cluster R est fixé à 35m, et puis la probabilité de devenir "cluster-head" $p = 5\%$. En suite, nous évaluons FRSM et CB-RSM avec leurs algorithmes de base "one-hop", notamment LEACH, LEACH-C, e-LEACH. Les performances sont évaluées par les trois métriques suivantes : la durée de vie, l'énergie restante globale et la quantité de données reçues à la station de base.

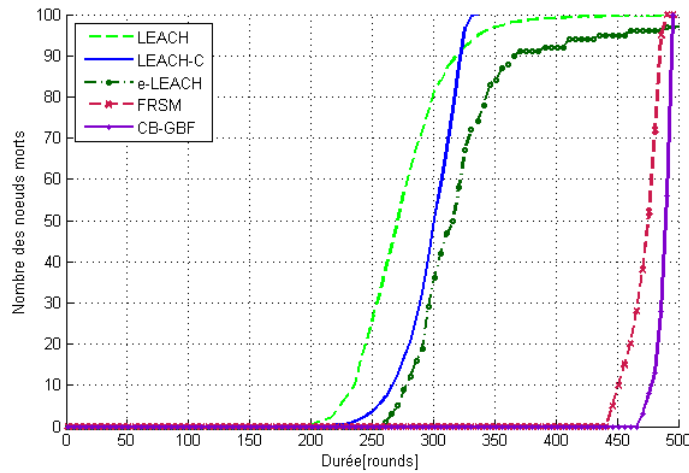


FIGURE 3.20 – Comparaison de la durée de vie : FRSM et CB-BGF

La figure 3.20 présente le nombre de nœud morts en fonction du temps (rounds) pour les cinq algorithmes de routage. Les résultats de cette figure montrent considérablement que FRSM et CB-RSM ont des meilleures performances que les autres algorithmes. Cela vient du fait que dans les protocoles LEACH, LEACH-C ainsi qu'e-LEACH, tous les "cluster-heads" transmettent directement les données agrégées à la station de base. Les "cluster-heads" qui sont loin de la station de base utilisent donc une puissance de transmission élevée, ce qui implique rapidement épuiser leur énergie. Néanmoins, l'algorithme CB-RSM considère l'énergie restante du nœud au cours de routage entre "cluster-heads" vers la station de base ainsi que la sélection du "cluster-head", qui lui permet de surpasser l'algorithme de FRSM. Les trois métriques pour la durée de vie sont améliorées de 80% pour FRSM et CB-RSM que les protocoles one-hop.

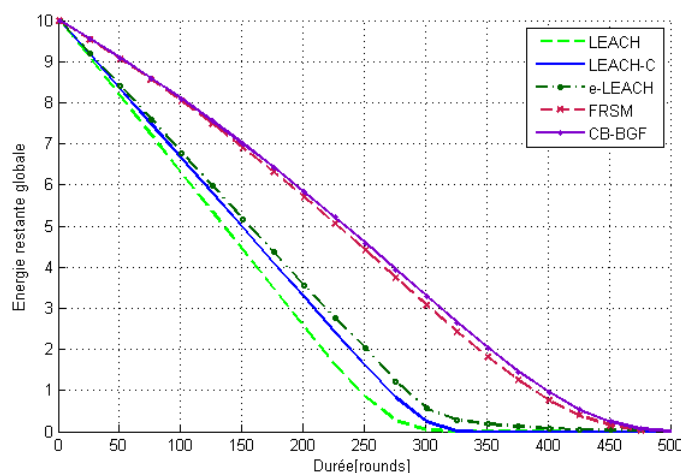


FIGURE 3.21 – Comparaison d'énergie restante : FRSM et CB-BGF

L'optimisation apportée par FRSM et CB-RSM est évidemment confirmée par les résultats de la figure 3.21. Cette figure démontre l'évolution d'énergie restante globale du réseau au cours du temps pour les cinq protocoles. Nous constatons clairement que les deux protocoles multi-hops améliorent beaucoup plus la dépense d'énergie que les protocoles one-hop. Cela vient par le fait que FRSM et CB-RSM allègent le problème de communication directe entre cluster-head et la station de base. Lorsque les "cluster-heads" sont élus, celui qui est plus loin de la station de base choisit dynamiquement le "cluster-head" successeur, auquel il a délivré les données agrégées, en admettant la construction d'un chemin élémentaire. Ainsi, les "cluster-heads" acheminent progressivement les données vers la station de base permettant d'ajuster la puissance de transmission en fonction de la distance de cluster-head voisin. Tous cela permettent d'équilibrer la consommation d'énergie à travers l'ensemble des "cluster-heads". Par conséquent, la CB-RSM et FRSM présentent une réduction de 80% de la consommation d'énergie par rapport au LEACH et e-LEACH.

La quantité de données reçues par la station de base pour les cinq protocoles est illustrée dans la figure 3.22. Cette figure présente le nombre de données cumulé reçus par la station de base en fonction du nombre de noeud survivant dans le réseau. Les résultats exposés dans cette figure démontrent l'efficacité de RFSM et CB-RSM en livrant plus de données que les autres. Bien que CB-RSM présente meilleure performance que FRSM. Cela est due au fait que FRSM considère l'agrégation de donnée dans la communication inter-cluster-head. Tandis que dans CB-RSM, lorsque le "cluster-head" reçoit des données à partir d'un autre "cluster-head", il transmettra les données directement au autre "cluster-head" relais sans agrégation

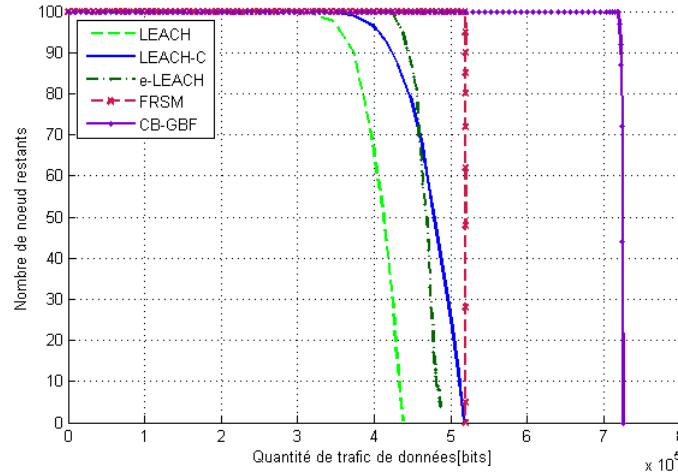


FIGURE 3.22 – Comparaison des traffics au sein du réseau :FRSM et CB-BGF

3.7.5 Conclusion

Dans cette section, nous avons proposé deux protocoles de routage hiérarchique basé sur clustering, dénoté FRSM et CB-RSM. Ces protocoles utilisent l'algorithme de clustering probabiliste discuté précédemment notamment LEACH et e-LEACH, respectivement. Ces protocoles forment des chemins de routage cluster-head en cluster-head afin d'effectuer la communication inter-clusters en se basant sur la construction de chemin élémentaire. FRSM utilise la puissance de la station de base pour interroger les CHs plus loin d'elle pour router leurs données. Tandis que CB-RSM utilise une technique d'auto-ordonnancement pour rendre les réseaux plus scalable. Les résultats des simulations ont démontré les performances de ces deux protocoles par rapport LEACH et e-LEACH. Ces gains des performances de FRSM et CB-RSM augmente d'environ 80% en termes de la durée de vie, d'efficacité énergétique ainsi que le nombre de paquet délivré à la station de base.

3.8 Conclusion de chapitre

Dans ce chapitre, nous avons développé et simulé des différents types des protocoles hiérarchique basé sur clustering, notamment centralisé et distribué, que ce soit "one-hop" ou "multi-hop". Compte tenu ces différents résultats, pour le réseau utilisant la technique de communication directe dans la phase de communication intra-cluster et inter-cluster, la méthode centralisée est plus performante en terme de durée de vie et d'efficacité énergétique dû à l'utilisation de la puissance de traitement de la station de base ainsi que la considération de l'énergie globale du ré-

seau dans le processus de sélection du "cluster-head". Cette méthode nécessite un échange périodique entre les noeuds et la station de base, cette étape va générer une augmentation de trafic de contrôle au sein du réseau, d'où une latence ainsi qu'une limitation au changement d'échelle. Le protocole distribué a été proposé pour pallier à ces problèmes. Lors de cette étude, nous avons constaté que la durée de vie pour les deux métriques HNA et LND augmente lorsque le nombre de clusters diminue. En revanche, la durée de vie augmente pour FND lorsque le nombre de clusters augmente, cela implique quasiment que tous les noeuds épuisent simultanément leur énergie.

Ces algorithmes de clustering proposés se basent sur l'optimisation de sélection de "cluster-head" parmi les noeuds ayant une réserve d'énergie élevée et une faible variance, et avec une rotation des noeuds dans le rôle du "cluster-head" de manière périodique afin d'améliorer la durée de vie du réseau. Cependant, le problème de la consommation déséquilibrée d'énergie perdure dû à la longue distance de transmission. Ainsi, la réduction de la distance de transmission des noeuds entre la station de base minimise la dissipation d'énergie. Néanmoins, l'auto-organisation et l'auto-ordonnancement sont des mécanismes qu'on ne peut éviter dans la mise en place d'une communication multi-saut lorsqu'on vise le passage à l'échelle du réseau. Aussi, nous avons proposé la technique de routage "greedy", appelé Binary Greedy Forwarding. Elle a pour objectif la minimisation de la consommation énergétique, le passage à l'échelle, augmentation de quantité de données livrées ainsi que la réduction de latence (complexité) dans la communication inter-cluster.

Caractérisation des consommations d'énergie d'un noeud

Sommaire

4.1	Introduction	89
4.2	Plateforme utilisée pour l'implémentation	90
4.2.1	Le système d'exploitation	90
4.2.2	Les noeuds du réseau	91
4.3	Méthodologie expérimentale de mesure des consommations	92
4.3.1	Les blocs des composants fonctionnels	93
4.3.2	Systèmes de mesure des consommations	96
4.4	Les résultats de mesures des consommations	102
4.4.1	Le microcontrôleur	102
4.4.2	Le module Timer	104
4.4.3	Les Leds (diodes électroluminescentes)	105
4.4.4	Le module radio	106
4.4.5	La mémoire flash externe	111
4.5	Modélisation et synthèse	111
4.5.1	Le microcontrôleur	112
4.5.2	Le transceiver	113
4.5.3	Synthèse des énergies consommées	117
4.6	Conclusions	121

4.1 Introduction

Comme nous l'avons vu dans le chapitre précédent, le modèle de consommation énergétique du noeud joue un rôle prépondérant dans l'évaluation de la durée de vie du réseau. Pour le moment le modèle le plus populaire (proposé dans [56]), a été

utilisé pour évaluer nos protocoles. Cependant, ce modèle a été obtenu uniquement à partir d'un modèle théorique simplifié du module de communication "radio".

Cela nous amène à envisager des questions qui font l'objet de ce chapitre. En effet, des mesures énergétiques préliminaires, nous ont révélé des divergences par rapport au modèle utilisé. Dans ce chapitre, nous ferons, en autres, une évaluation énergétique de la consommation d'un noeud à l'aide des mesures de consommations réelles des différents composants qui interviennent dans les processus de fonctionnement de celui-ci.

Afin de pouvoir mener à bien les travaux de mesures, nous décrirons les étapes de mesure de consommation d'énergie de chaque phase. Ainsi deux étapes principales sont incontournables dans cette approche : l'implémentation des codes de programme dans les noeuds et l'instrumentation des circuits de mesure.

Ce chapitre sera organisé comme suit : dans la section 4.2, nous présentons brièvement les plateformes utilisées pour notre mesure. Notamment le système d'exploitation du noeud de réseau et le noeud nécessaire pour la mise en oeuvre de nos expériences. Ensuite, nous décrirons dans la section 4.3 la méthodologie adoptée afin d'obtenir des résultats expérimentaux fiables et valides. La section 4.4 sera consacrée aux caractérisations des résultats issus des tests. Elle présentera également une synthèse de ces résultats et les éléments de corrélation ayant permis de mettre en place un nouveau modèle énergétique. Enfin, nous conclurons ce chapitre par la section 4.6.

4.2 Plateforme utilisée pour l'implémentation

Nous commençons cette section par la présentation succincte de l'OS TinyOS [136] : un système d'exploitation typiquement utilisé pour programmer des noeuds de réseaux sans fil. Ensuite nous présenterons brièvement les cartes TelosB [1] qui ont été utilisées pour nos expériences afin d'évaluer les consommations énergétiques.

4.2.1 Le système d'exploitation

TinyOS est un système d'exploitation intégré, modulaire, open-source pour les réseaux de capteurs sans fil conçu par l'université américaine de Berkeley. Ce système d'exploitation a acquis rapidement un succès international en devenant une solution de référence pour l'implémentation et l'évaluation de WSNs [137]. Sa conception a été entièrement réalisée en NesC, langage orienté composant, qui se rapproche syntaxiquement du langage C [136, 138]. TinyOS a été créé pour répondre aux caractéristiques et aux nécessités des réseaux de capteurs, telles que :

- une taille de mémoire réduite,
- une basse consommation d'énergie,

- les opérations robustes,
- une optimisation en terme d'usage de mémoire et d'énergie.

Le plus gros avantage de TinyOS consiste au fait qu'il est basé sur un fonctionnement événementiel, c'est-à-dire qu'il ne devient actif qu'à l'apparition de certains événements. Le reste du temps, le capteur se trouve en état de veille afin de garantir une durée de vie maximale par rapport aux faibles ressources énergétiques du capteur. TinyOS se distingue aussi par son caractère non préemptif, c'est-à-dire qu'il ne gère pas les interruptions entre tâches (une tâche ne peut pas interrompre une autre tâche). Par contre, il donne une priorité aux interruptions matérielles qui peuvent à tout moment stopper l'exécution d'une tâche (i.e., les événements peuvent interrompre les tâches). L'implémentation des composants (programmes) dans TinyOS est donc basée sur une structure à trois niveaux de planification :

- les commandes : elles sont l'exécution d'une fonctionnalité précise dans un autre composant. Exemple : l'activation d'un Timer.
- Les événements : ils sont l'équivalent logiciel d'une interruption matérielle. De plus, ils peuvent interrompre les tâches qui sont exécutées. Exemple : quand le compteur du " Timer " arrive à son terme.
- Les tâches : les tâches ont pour rôle de réaliser une plus grande quantité de traitement et elles ne sont pas critiques dans le temps. Les tâches sont exécutées complètement, mais l'initialisation et la terminaison d'une tâche sont des fonctions séparées. Les tâches ne peuvent pas prendre de paramètres en entrée.

A l'appel d'une tâche, celle-ci va prendre place dans une file d'attente de type FIFO pour être exécutée. Une tâche activée s'exécute entièrement car il n'y a pas de mécanisme de préemption entre les tâches. Lorsque la file est vide, le système met en veille le dispositif jusqu'au lancement de la prochaine interruption. Les événements sont prioritaires par rapport aux tâches et peuvent interrompre les tâches en cours d'exécution. Ils permettent de faire le lien avec les interruptions matérielles.

4.2.2 Les noeuds du réseau

Les noeuds telos sont les noeuds de réseau les plus utilisés actuellement dans le domaine de la recherche et l'évaluation de solutions protocolaires pour les RSCFs. Il existe deux types de noeuds : TelosA et TelosB [26, 139, 140]. TelosB utilise le microcontrôleur MSP430 F1611, qui remplace le microcontrôleur F149 de la révision A. Nous avons implémenté notre code nesC sur les cartes TelosB afin de mesurer les différentes consommations des composants. La Figure 4.1 nous présente une carte TelosB et ses différents composants.

Ces caractéristiques peuvent globalement se résumer comme suit :

- Un microcontrôleur MSP430 de Texas Instrument cadencé à 8 MHz, qui dis-

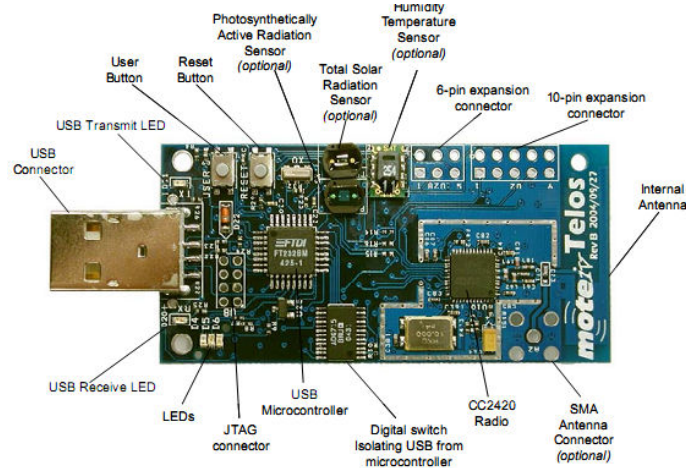


FIGURE 4.1 – Noeud de réseau telosB [1]

pose de mémoire vive 10 KB (RAM) et de 48 KB de mémoire non volatile (ROM). Le MSP430 dispose aussi que des convertisseurs analogique-digital et un contrôleur DMA intégrés.

- Un module Radio Chipcon (Transceiver) CC2420 travaillant sur une bande de fréquence radio s'étendant de 2.4 GHz à 2.4835 GHz, ayant un débit théorique de transmission 250 kbps, respectant la norme IEEE 802.15.4.
- Une liaison USB, via laquelle la carte peut se connecter à un ordinateur par exemple. Son utilisation ne demande pas l'installation de logiciel spécifique, la carte apparaît comme un port série au niveau de l'OS.
- Un bouton utilisateur dédié à des besoins spécifiés dans le programme de l'utilisateur. Il est possible de déclencher l'exécution d'une commande en appuyant sur ce bouton (ex : commencer le début d'un processus de fonctionnement).
- Très faible consommation électrique.
- Temps de réveil inférieur à 6 microsecondes.
- Un support d'expansion de 16 ports d'entrées/sorties.
- Une carte supportée par TinyOS.

4.3 Méthodologie expérimentale de mesure des consommations

Cette section fournit une description détaillée de la méthodologie expérimentale, y compris l'identification des modules fonctionnels élémentaires, la méthode de mesure et le système de mesure utilisé.

4.3.1 Les blocs des composants fonctionnels

Le système doit être subdivisé en des composants fonctionnels principaux afin d'effectuer des mesures de consommation d'énergie valides et exploitables. Cette approche permet en outre de déterminer plus précisément la consommation d'énergie de chacun d'eux. Le processus d'évaluation est basé essentiellement sur la structure matérielle du TelosB (Fig.4.1, Fig.4.20), tout en suivant un protocole valide comme par exemple, l'identification séparée et séquentielle des modules qui constituent les différentes étapes pour envoyer ou recevoir des données. De ce fait, la procédure de mesure suivie pour chaque module fonctionnel dépend aussi du fonctionnement interdépendant des modules. Il faudra considérer ce point pour toutes les mesures possibles. Cette méthode de mesure demande des techniques pour accéder aux composants matériels, par le biais de codes simples qui sont développées, et s'appuient sur la couche HAL (Hardware Adaptation Layer) construite sur le modèle HAA (Hardware Abstraction Architecture) dans TinyOS 2.0 [141,142]

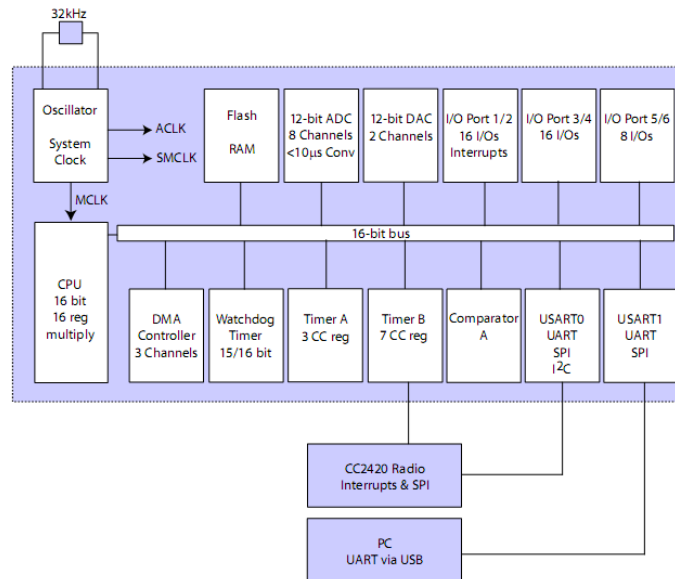


FIGURE 4.2 – Schéma synoptique de TI MSP430, Telob [1]

L'objectif principal de cette méthode est de désactiver tous les composants inutiles, et de permettre d'initialiser seulement les modules fonctionnels nécessaires. Ceci se fait avec l'aide de l'HAL défini, pour une opération donnée en cours d'évaluation. L'initialisation est suivie par la configuration de paramètres spécifiques en respectant l'opération de la mesure, notamment la fréquence du processeur, le mode de faible niveau de puissance, etc...

Nous pouvons ainsi par cette méthodologie réaliser :

1. un noeud opérationnel couvrant entièrement la fonctionnalité spécifique en cours d'évaluation,
2. l'isolement de tous les modules non indispensables mais consommateurs de d'énergie,
3. et l'enregistrement de l'opération complexe comme une séquence de ses fonctionnalités élémentaires.

Sur la base de cette approche, des mesures précises et réalistes sont extraites afin de composer le modèle de la consommation d'énergie liée au réseau étudié.

4.3.1.1 Le module microcontrôleur

Commençons par le microcontrôleur d'un noeud capteur. Il est au coeur de l'architecture matérielle. Rappelons que TelosB utilise un microcontrôleur msp430F1xx 16 bits pour piloter tous les autres composants du noeud (Fig.4.20). Ce microcontrôleur est réputé pour sa très faible consommation énergétique. Comme caractéristiques, il dispose en plus de ce qui a été présenté de 128 B de stockage d'information. Il possède également un système d'horloges flexible conçu spécialement pour une utilisation avec une batterie [143, p.1-2]. Ce système d'horloges met à disposition de l'utilisateur trois sources d'horloge différentes en fonction des besoins en consommation et en fréquence :

1. ACLK (Auxillary clock) : utilisée pour les périphériques.
2. MCLK (Master clock) : utilisée par la CPU et pour les périphériques grande vitesse.
3. SMCLK (Sub main clock) : également utilisée pour les périphériques.

Typiquement SMCLK fonctionne à la même fréquence que MCLK, dans la gamme des mégahertz. L'ACLK est souvent issue d'un quartz. Tandis que, MCLK et SMCLK sont fournies par l'*oscillateur à commande numérique* (DCO : Digitally Controlled Oscillator). Ce dernier est commandé par une *boucle à verrouillage de fréquence* (FLL : Frequency-Locked Loop). L'ACLK est souvent utilisée pour un *RTC background* dans la fonction d'auto-réveil. En effet, la plupart des périphériques peuvent sélectionner soit la SMCLK, soit l'ACLK. Les sources d'horloge disponibles varient considérablement entre les familles et les périphériques individuels.

Le MSP430 possède aussi deux Timers internes configurables indépendamment (Timer A et Timer B), un module ADC 12 bits et un contrôleur DMA [143, 144].

Les différents états de fonctionnement du microcontrôleur MSP430 doivent aussi être considérés dans notre approche. En effet, MSP430 possède 6 différents états de puissance en fonction de la configuration active de ses systèmes d'horloge. Il est à noter que cette configuration est initialisée en fonction de l'interdépendance de tous les autres sous-systèmes et leurs périphériques associés nécessaires. Le microcontrôleur doit toujours être dans l'état le plus économique en termes d'énergie

tout en satisfaisant aux exigences de l'application. Pour le noeud TelosB, chaque fois qu'un microcontrôleur traite une interruption, il se déplace d'un mode de faible puissance à un mode actif. Et chaque fois que l'ordonnanceur TinyOS trouve la file d'attente FIFO de tâches vides, il renvoie le microcontrôleur à un état de faible puissance [1, 139].

Pour le module ADC 12-bits : il permet de convertir les signaux analogiques mesurés par le capteur en signaux numériques afin qu'ils puissent être traités par la partie logicielle du noeud. Cette chaîne de conversion est aussi constituée d'un contrôleur et d'une partie mémoire pour le stockage. Le nombre de conversions à effectuer est configurable avant le lancement de l'interruption de fin de conversion, de même pour le nombre de cycles d'horloge avant la prochaine série de conversions. L'ADC contient un tampon de 16 entrées de 12 bits permettant de convertir les données et de stocker les résultats sans avoir à passer par le CPU [144].

Pour le contrôleur DMA : il permet de transférer des données d'une adresse mémoire à une autre sans intervention du CPU. Les avantages d'utilisation du DMA résident sur les transferts plus rapides (ils ne nécessitent que deux cycles d'horloge) et sur la minimisation d'occupation du CPU. Cela permet un fonctionnement dans un mode de faible consommation et surtout de libérer également le CPU pour effectuer d'autres tâches.

4.3.1.2 Le module Timer

Les Timers jouent le rôle de temporisateur permettant d'émettre une interruption aux autres modules externes comme le module radio, capteur, etc... Dans Tinyos, le module Timer est fréquemment utilisé, pour enclencher ou déclencher un événement. Une caractéristique à bien avoir en mémoire est la caractéristique d'énergie consommée du Timer en fonction de sa fréquence d'horloge et sa période.

4.3.1.3 Le module Radio

Il s'agit d'un bloc de communication sans fil appelé transceiver dans la documentation technique. Il permet de transmettre les données acquises aux autres entités communicantes. Ses performances sont conditionnées par la couche physique implémentée. Elles dépendent donc de la bande de fréquence utilisée, du niveau de puissance d'émission, de la portée de transmission, du type de modulation, de la présence ou non de codes correcteurs d'erreurs, et enfin du niveau de consommation. Le module Telosb est équipé d'un transceiver CC2420 Chipcon pour établir la communication [2, 139]. Ce dispositif est contrôlé par le microcontrôleur MSP430 par l'intermédiaire d'un port SPI ainsi que par une série de lignes et les interruptions d'E/S numériques. Le microcontrôleur peut éteindre le CC2420 en cas de besoin. Le module CC2420 est aussi caractérisé par sa capacité à détecter le niveau de

signal reçu RSSI (Received Signal Strength Indication) et aussi à gérer le canal de communication (CCA : Clear Channel Assessment). Ce dernier est utilisé pour l'implémentation de protocoles MAC type CSMA [30].

Pour estimer la puissance consommée du module radio, on considère les états principaux suivants : mode émission, mode réception, On/Off, et écoute. Les mesures de la consommation de puissance d'émission et de réception seront effectuées indépendamment. L'objectif est d'évaluer la consommation d'énergie pendant la phase transmission en variant la puissance d'émission et la taille de paquets. L'évaluation de la phase réception sera faite en fonction de la taille du paquet.

4.3.1.4 Le module mémoire

Plusieurs types de mémoires sont nécessaires au fonctionnement d'un noeud. Une mémoire de type RAM (Random Access Memory) est nécessaire pour stocker les données temporaires acquises par le capteur, mais également les paquets issus des autres noeuds. Malgré sa rapidité, ce type de mémoire possède l'inconvénient de perdre son contenu si l'alimentation est interrompue. Une mémoire de type ROM (Read Only Memory) est nécessaire pour stocker le code des programmes et l'OS. Ces deux mémoires sont déjà incorporées dans le microcontrôleur. Cependant, une mémoire FLASH peut également servir pour stocker temporairement les données provenant de la RAM en cas d'espace insuffisant de cette dernière ou en cas de coupure d'alimentation. C'est pourquoi, la famille du noeud telos utilise aussi ce type de mémoire FLASH. Malgré ses intérêts, la mémoire FLASH, du fait de ses temps d'accès en écriture ou en lecture, est une source de consommation d'énergie assez importante [145]. Un des objectifs de mesure sera aussi d'évaluer la consommation d'énergie de cette mémoire Flash pendant la phase d'écriture et lecture de données.

Voyons maintenant à proprement parlé les méthodes de mesure et de caractérisation de ces éléments.

4.3.2 Systèmes de mesure des consommations

Il existe deux méthodes pour mesurer la consommation énergétique des composants du noeud : la méthode interne et la méthode externe. Pour la méthode interne, TelosB possède un contrôleur batterie qui permet d'interroger celle-ci afin de connaître le niveau de charge actuel et d'en déduire la puissance dissipée instantanée. Pour la méthode externe, une résistance est ajoutée dans le circuit d'alimentation du noeud afin d'avoir l'image du courant. On peut alors calculer la puissance dissipée dans le circuit dont on souhaite avoir la consommation.

Pour la méthode interne, peu de matériel supplémentaire est nécessaire tant que le système dispose d'un contrôleur batterie intelligent. Cependant, les mesures effectuées concernent le fonctionnement du noeud dans son intégralité. Il faut donc avoir

des valeurs de référence des mesures de chaque sous-système indépendamment, en se référant à la datasheet de chaque composant qui caractérise les noeuds. Ce qui donnera par la suite la consommation du système à vide. Ces mesures et cette méthode ne sont pas très précises pour la définition d'un modèle proche du fonctionnement réel du noeud.

La méthode externe demande pour sa part des appareils et composants additionnels : au moins une résistance de haute précision à insérer dans le circuit et un voltmètre numérique précis haute fréquence. Il faut aussi considérer la question des relevés des résultats, du stockage des fichiers de mesures et leur traitement. Pour avoir des mesures fiables, il faut d'autres appareils supplémentaires, notamment un ordinateur, un logiciel de gestion des instruments et un logiciel de traitement de données. En effet, les mesures de la tension aux bornes de la résistance sont nécessaires pour avoir l'image du courant traversant le circuit afin de finalement déterminer la puissance dissipée instantanée du système. Ensuite, il faut en extraire les parties correspondantes aux événements que l'on souhaite analyser pour pouvoir obtenir des résultats ayant des significations réelles. Malgré tous ces inconvénients, les résultats apportés par la méthode externe sont fiables, tout en permettant de mesurer des différents composants actifs selon le fonctionnement actuel du noeud.

4.3.2.1 La méthode externe adoptée

Pour notre système de mesure, nous avons choisi la méthode externe pour sa précision bien supérieure à la méthode interne. Afin de mesurer la consommation d'énergie d'un circuit électronique, nous allons substituer un générateur de tension au circuit d'alimentation du noeud (les piles). Par la suite une résistance shunt est reliée en série avec le noeud TelosB et le générateur (Fig. 4.3). La mesure différentielle aux bornes de la résistance apporte une sensibilité au bruit très réduite.

Deux configurations sont possibles dans cette approche. La résistance peut être positionnée entre la partie positive de l'alimentation ou dans la partie négative (masse). La première est appelée "high-side current sensing" tandis que l'autre "low-side current sensing". Cette dernière nous a paru acceptable parce que la tension différentielle aux bornes de la résistance est égale à la tension mesurée par rapport à la masse et la connexion entre la résistance et le noeud. Cela simplifie l'amplification parce qu'il n'y a pas de tension en mode commun. Par contre, cette technique crée un problème de fluctuation de masse lorsque le courant varie. Ce qui est indésirable du fait que de nombreux composants électroniques sont sensibles aux variations de masse. De plus la résistance doit être maintenue faible, car l'ampleur de cette fluctuation est proportionnelle à la résistance. En plaçant la résistance sur le côté positif, les fluctuations de tension seront décalées vers le côté positif. Ceci est plus favorable, car la plupart des composants sont plus résistants aux fluctuations de la

borne positive. Nous avons adopté cette dernière configuration.

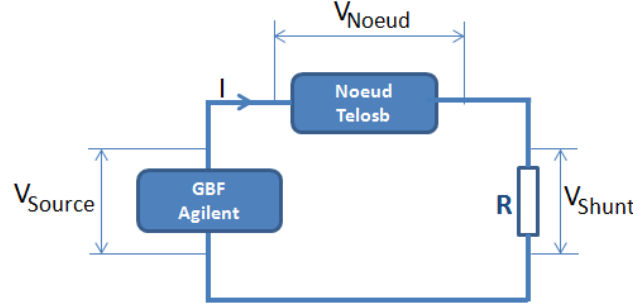


FIGURE 4.3 – Mesure d'énergie avec la résistance shunt

Grâce à un calcul sur la loi d'Ohm, nous déduisons la valeur de l'intensité du courant qui traverse le noeud, $I = \frac{V_{Shunt}}{R_{Shunt}}$. De cette façon, l'expression 4.1 nous permet de déterminer la puissance consommée instantanée par le noeud TelosB à chaque instant t .

$$P_N(t) = V_{Noeud} * I(t) \quad (4.1)$$

Puis, nous procédons à l'intégration de cette puissance pour connaître l'énergie consommée, $E(t_1, t_2)$, associée à un intervalle de temps $[t_1 ; t_2]$:

$$E(t_1, t_2) = \int_{t_1}^{t_2} P_N(t) dt \quad (4.2)$$

Le courant consommé du noeud Telosb est très variable : de quelques milliam-pères au microampères selon l'activité du noeud [1]. Les premières mesures s'avèrent très sensibles au bruit. Il est nécessaire d'utiliser un circuit permettant d'amplifier les signaux mesurés et de minimiser en même temps les interférences. D'où la nécessité d'un amplificateur d'instrumentation. L'un des avantages de l'amplificateur d'instrumentation est qu'il dispose un bloc de gain réglable en boucle fermée qui a une entrée différentielle. Sa sortie qui est asymétrique par rapport à une borne de référence. De plus, la réjection en mode commun est importante. Elle permet d'annuler tous les signaux indésirables (le même potentiel sur les deux entrées), tandis que l'amplification des signaux est différentielle (une différence de potentiel entre les entrées de l'amplificateur) [146]. Le gain différentiel sera noté G .

Nous avons donc utilisé l'AD627 qui est un amplificateur d'instrumentation de haute précision [147]. En branchant une résistance en série avec le noeud capteur, la puissance instantanée est ainsi accessible : le courant peut être multiplié par la tension provenant d'un canal de mesure séparé. Et en divisant par le gain différentiel G nous pouvons avoir la véritable consommation d'énergie (équ.4.3).

$$P_N(t) = (V_{Source} - V_{Shunt}) * \frac{V_{Shunt}}{R_{Shunt}} * \frac{1}{G} \quad (4.3)$$

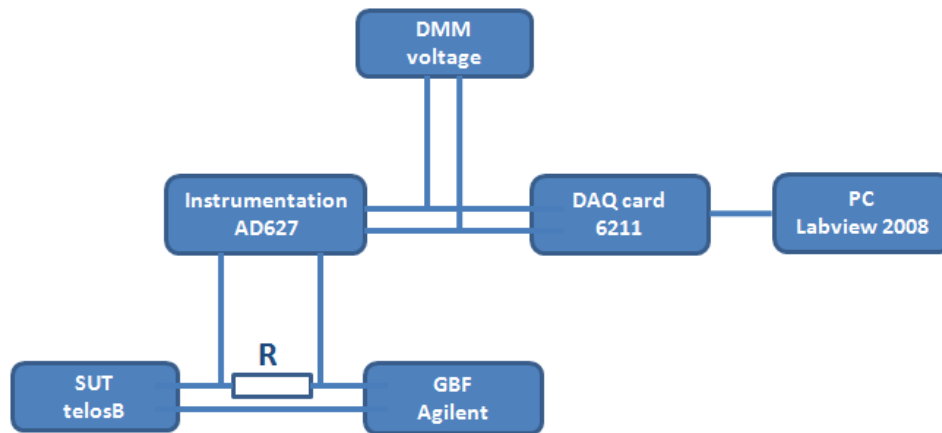


FIGURE 4.4 – schéma synoptique de mesure

Dans notre expérience, une résistance shunt de $12\ \Omega$ a été insérée en série entre l'alimentation (GBF Agilent) et le noeud. Les deux extrémités de la résistance sont reliées aux entrées d'un amplificateur d'instrumentation avec un gain de 10 pour des courants de l'ordre du milliampère [mA] et un gain de 100 pour des courants de l'ordre du micro-ampère [μA]. La sortie de ce circuit est reliée à une voltmètre numérique (noté DMM sur la Fig.4.4) pour nous permettre de vérifier à tout moment la tension aux bornes de ce circuit. La tension de sortie est aussi connectée à un boîtier d'acquisition pour numériser les signaux avec précision. Cette chaîne d'instrumentation facilite le stockage de données ainsi que l'automatisation de traitement sous un logiciel dédié (Fig.4.4).

La fréquence d'échantillonnage que nous avons choisie nous permet de mesurer précisément la consommation énergétique et d'identifier facilement les différentes étapes de fonctionnement du noeud. Nous avons adopté une fréquence de 50 kHz soit 50000 échantillons par seconde. Pour moyenniser les mesures, en laissant le système acquérir les données pendant 10 secondes, le nombre de points d'une grandeur étudiée s'élève à 500000, soit environ un fichier de mesure de 40 Mo. L'acquisition doit aussi commencer juste avant l'évènement à caractériser. Nous avons traité ce problème en développant un ensemble d'outils logiciels associés à un interrupteur : un outil logiciel au niveau de telos permet de tenir compte de l'évènement associé au bouton externe pour démarrer le système. A partir des signaux amplifiés par l'AD627, nous avons aussi pu développer un outil supplémentaire au niveau acquisition : cette fonction permettant de détecter facilement les fronts montant et descendant des signaux. Cette approche nous permet d'accéder aux durées et aux calculs mathématiques associés.

Pour fiabiliser la mesure, l'AD627 nécessite une étape de calibrage. Les documents constructeurs [147] ne fournissent pas d'informations précises pour l'utilisa-

tion fiable de ce composant.

4.3.2.2 Calibrage de l'AD627

En se basant sur notre plate-forme, nous avons envisagé de réaliser un circuit de calibrage simple qui n'a pas d'alimentation symétrique. La tension de référence de l'AD627 est reliée directement à la masse.

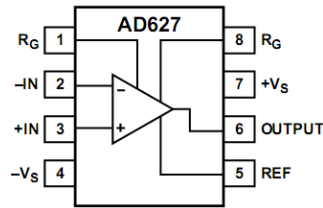


FIGURE 4.5 – Brochage de l'AD627

La figure (4.5) nous montre la numérotation et l'indication des brochages de l'AD627. Pour régler le gain souhaité, une résistance doit être branchée entre les broches 1 et 8 de l'amplificateur. De ce fait, l'utilisateur peut calculer la valeur souhaitée de la résistance pour un gain donné (Equ.4.4), ou lire directement les résultats de calcul de gain répertoriés dans la fiche technique [147].

$$R_G = \frac{200k\Omega}{Gain - 5} \quad (4.4)$$

Pratiquement, nous pouvons ainsi choisir la résistance normalisée insérée au circuit afin d'en déduire la valeur approximative du gain souhaité à partir de l'équation (4.4).

Après avoir choisi le gain, on a procédé à un test avec la configuration de l'AD627 souhaité en mesurant la tension aux bornes de la résistance shunt et à la sortie de l'AD627. Nous avons constaté que les résultats obtenus n'étaient pas cohérents.

L'incertitude de mesure INC totale à la sortie de mesure peut être déterminée comme suit, avec un gain G fixé à 10 :

$$\begin{aligned} INC_{mesure} &= Gain \times V_{OSI} + V_{OSO} \\ &= 10 \times 250 + 1000 \\ &= 3500 \\ &= 3.5[mV] \end{aligned}$$

Avec deux erreurs de tension d'offset à considérer : l'offset d'entrée V_{OSI} , et l'offset de sortie V_{OSO} . Pour l'AD627, en alimentation simple, la fiche technique indique que $V_{OSI} = 250[uV]$ et $V_{OSO} = 1000[uV]$ ([147]).

Dans notre cas d'études, on ne peut pas négliger cette erreur, vu que notre plage de mesure est de l'ordre du milliampère voir même du microampère.

Nous avons alors pris la décision d'ajuster la tension d'offset qui devrait être calculée ou enlevée afin d'avoir des mesures correctes.

Dans la pratique, pour ajuster l'offset de l'amplificateur, nous devons mettre à la masse les deux entrées ($-IN$ et $+IN$) et mesurer la sortie, et le résultat devrait être comptabilisé pour toutes les mesures. Toutefois, lorsque les deux entrées sont à la masse, nous n'avons relevé aucune tension d'offset dans nos mesures (Fig.4.6).

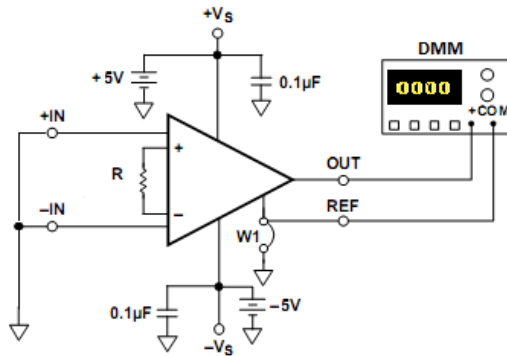


FIGURE 4.6 – Montage pour relever de l'offset de tension

Nous avons alors regardé le courant de polarisation (Input Bias Current) et le courant d'offset (Input Offset Current) d'entrée comme sources potentielles d'erreur. Le courant Input-Bias sur les entrées est le courant qui circule (entrant ou sortant), aux deux bornes d'entrée de l'amplificateur d'instrumentation [146]. Le courant d'offset de l'entrée est la différence entre les deux courants input-bias d'entrée et cela entraîne des erreurs d'offset dans l'amplificateur d'instrumentation lorsque les résistances d'entrée dans deux bornes d'entrées sont différentes. Les courants Input-Bias peuvent être considérés comme une source de tension d'offset (en effet, un courant qui circule à travers une résistance de source provoque une tension).

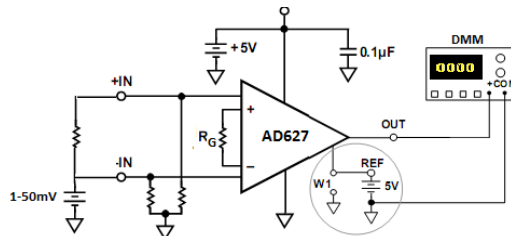


FIGURE 4.7 – Circuit pour le calibrage par minimisation de courant d'offset - ajout de résistance d'entrée.

102 Chapitre 4. Caractérisation des consommations d'énergie d'un noeud

Comme solution à ce problème, deux résistances sont branchées entre les deux entrées de l'amplificateur d'instrumentation et la masse. Ces résistances doivent être dimensionnées attentivement pour réduire l'offset et le CMRR erreur. Nous avons effectué plusieurs mesures en utilisant des résistances de différentes valeurs ($10k\Omega$ à $10M\Omega$) Fig.4.7. Nous avons au final retenu les résultats obtenus en branchant une résistance de $1M\Omega$ entre chaque entrée et la masse pour résoudre ce problème. Pour la suite de notre expérimentation, le circuit finalisé est décrit à la Fig.4.8.

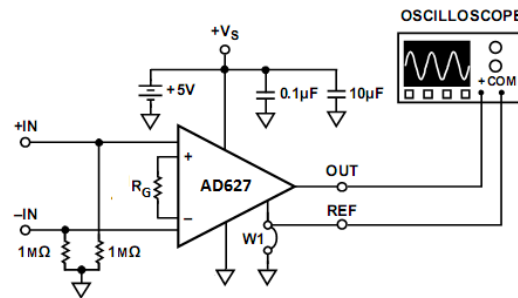


FIGURE 4.8 – Schéma de mesure final à base de l'AD627 pour les mesures de consommation.

Dans notre cas d'étude, le courant consommé de notre système varie entre $180\mu A$ et $40mA$. Avec une résistance de shunt 12Ω , la différence de potentiel présente aux bornes d'entrée de l'AD627 sera entre $2.16mV$ et $480mV$. La chaîne d'instrumentation étant décrite, voyons à présent les résultats de mesure.

4.4 Les résultats de mesures des consommations

L'objectif de cette section est de présenter les résultats de mesure des consommations obtenues en respectant la méthodologie décrite dans la section précédente. Nous identifierons les différentes parties de chaque module afin de proposer un modèle énergétique significatif.

4.4.1 Le microcontrôleur

Le microcontrôleur MSP430 s'avère assez souple en termes de consommation d'énergie. Cela est dû au mécanisme DPM (Dynamic Power Management), qui permet de désactiver dynamiquement les composants du noeud de capteur et de les réveiller en cas de besoin. Ce mode de gestion de puissance présente des transitions d'états. Ils sont représentés par des modes d'alimentation différents correspondant à l'arrêt du CPU, de la mémoire, des horloges ou des autres périphériques internes. Il est à noter que chaque transition de l'état prend un certain temps. Cela consomme

Mode	Description	Observations
LPMA	CPU, toutes les horloges et les modules sont actifs	Le MSP430 démarre dans ce mode lorsque le CPU est nécessaire. Pour l'activer, une interruption commute automatiquement l'appareil en mode actif.
LPM0	CPU et MCLK sont désactivés, DCO, SMCLK et ACLK sont actives.	Utilisé lorsque le CPU n'est pas nécessaire, mais certains modules nécessitent une horloge rapide de SMCLK et le DCO.
LPM1	CPU, MCLK, DCO osc. sont désactivées. DC generator est désactivée si le DCO n'est pas utilisé pour MCLK ou SMCLK en mode actif SMCLK, ACLK restent actifs	LPM1 et LPM2 sont rarement employés dans les dispositifs actuels [12].
LPM2	CPU, MCLK, SMCLK, DCO osc. sont désactivées Générateur de courant continu reste activé ACLK est actif	
LPM3	CPU, MCLK, SMCLK, DCO et son générateur de courant continu sont désactivés. Seule ACLK reste actif.	C'est le mode standard lorsque le dispositif doit se réveiller à intervalles réguliers . Il est également nécessaire si le MSP430 doit maintenir une horloge en temps réel.
LPM4	CPU et toutes les horloges sont désactivées.	Le dispositif ne peut être réveillé que par un signal externe. Il est également appelé mode de rétention de RAM.

TABLE 4.1: Description des Low Power Modes (LPM) du MSP430

par conséquent de l'énergie.

Dans chaque mode de puissance, appelé également Mode de Faible Puissance (Low Power Mode), différents périphériques sont progressivement éteints. Le MSP430 possède cinq modes de faible puissance configurables, à l'exception de l'état actif complet [144]. Chaque passage de l'état inactif à un LPM a un coût énergétique fixe, qui est généralement négligeable [144]. Cependant, le coût de l'énergie pour réveiller le microcontrôleur à partir d'un mode de faible puissance augmente avec la profondeur du mode de faible puissance considéré. Pour cette raison, il est important de réduire le nombre de transitions d'état en équilibrant convenablement la stratégie de mécanisme d'ordonnancement sans utiliser d'arrêt ou de démarrage brusque.

La mesure de la consommation d'énergie de chaque mode peut fournir des informations précieuses sur l'économie d'énergie attendue par l'utilisation du LPM appropriée. Le tableau 4.1 nous présente les différents modes avec les descriptions associées. Le mode de faible puissance varie de LPM0, qui désactive seulement le processeur et l'horloge principale MCLK, à LPM4, qui désactive le processeur, toutes les horloges, et l'oscillateur, jusqu'à ce qu'il soit réveillé par une source d'interruption externe.

La figure 4.9 compare les résultats de consommation de courant pour l'état actif (A) et les cinq modes LMP (0-4) pour une fréquence d'horloge de 4 MHz, et une tension d'alimentation constante de 3V. On constate que lors du passage de l'état actif

Mode	Courant Mesuré [mA]			Datasheet courant telosb		unité
	Max	Moyen	Min	Max [mA]	Min [mA]	
LPMA	2,18	2,05	1,99	2,4	1,8	mA
LPM0	185,65	179,65	178,32	N/A	N/A	μA
LPM1	185,28	180,69	178,68	N/A	N/A	μA
LPM2	42,49	42,15	41,67	N/A	N/A	μA
LPM3	20,47	19,84	19,01	N/A	N/A	μA
LPM4	14,75	14,21	13,76	21	5	μA

TABLE 4.2: Msp430 : Courants consommés comparés dans les différents LPMs.

à LPM0, la consommation d'énergie du microcontrôleur est déjà réduite d'environ 12 fois. En revanche, les consommations de deux modes LPM0 et LPM1 sont quasi-identiques. Cela est dû au fait que seule la boucle de contrôle de l'horloge MCLK est désactivée. Tandis que, celles du passage entre LPM1 à LPM2 et LPM3 sont réduites respectivement d'environ 4.5 et 5 fois. Telos a adopté comme configuration par défaut le mode LPM3. Mais s'il détecte que le TimerA, les USARTs, ou l'ADC sont actifs, il se met en mode LPM0.

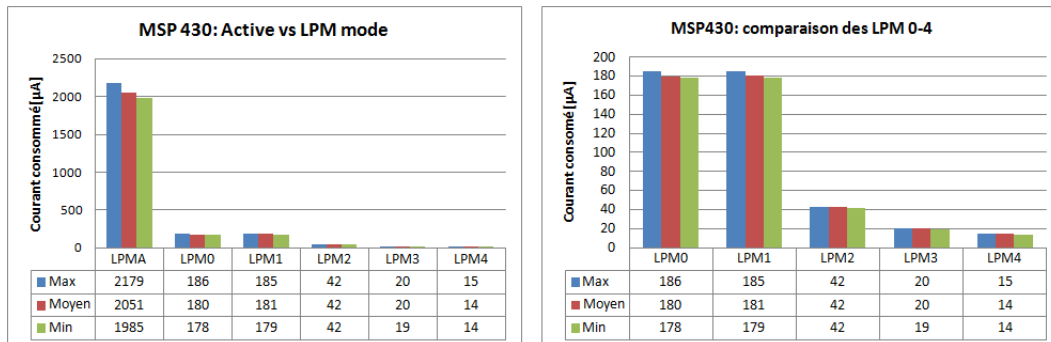


FIGURE 4.9 – Msp430 : Courants consommés dans les différents LPMs.

Les résultats présentés dans le tableau 4.2, sont des valeurs moyennes de 10 différentes mesures. Nos résultats confirment les limites supérieures et inférieures de la documentation constructeur du Telosb.

L'énergie totale consommée par le microcontrôleur pourra être déduite par la somme des produits de puissance dissipée par chaque état activé et sa durée d'activation, pour un protocole donné.

4.4.2 Le module Timer

Il existe deux différents modes d'opération de Timer : (A) Timer Périodique : le Timer se déclenche périodiquement après la première initialisation. Dans notre mesure, le Timer est déclenché toutes les secondes après la première initialisation.

(B) Timer One-shot (monostable) : ce Timer nécessite une réinitialisation après chaque enclenchement, soit toutes les secondes.

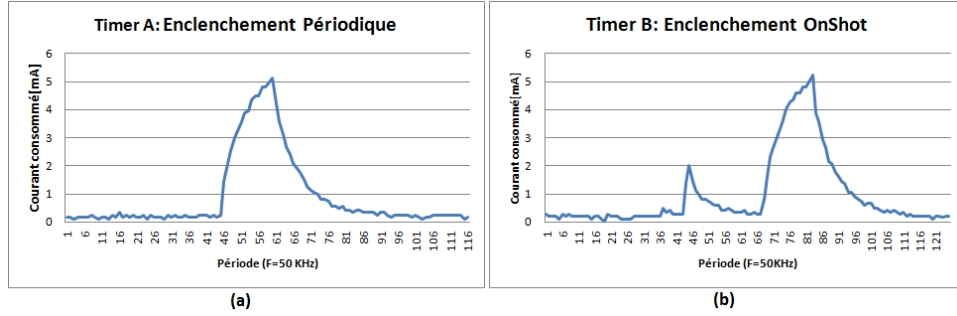


FIGURE 4.10 – Courant consommé : (a)Timer Périodique, (b)Timer One-shot

La figure 4.10 illustre les résultats des mesures effectués pour les deux modes de fonctionnement. Nous remarquons que le courant d'enclenchement de deux modes est quasi-similaire allant jusqu'à 5mA avec une durée de déclenchement de 900 μ s. A la différence près, le mode one-shot présente un autre appel de courant de 2mA avec une durée de déclenchement d'environ 450 μ s (Fig.4.10.b). En effet, la consommation du Timer est fixe à chaque événement. Par rapport à la consommation d'énergie du microcontrôleur, le Timer consomme environ 50% de plus, pour une même durée de fonctionnement (i.e : 900 μ s). Par conséquent, l'effet de la consommation du Timer dans la consommation globale n'est pas à négliger.

4.4.3 Les Leds (diodes électroluminescentes)

Les Leds existent dans toutes les plates-formes utilisées couramment et constituent une source importante de consommation d'énergie. Même si elles ne sont pas obligatoirement recommandées, elles sont utilisées pour contrôler visuellement le processus de fonctionnement du protocole évalué. Le comportement énergétique de ces éléments a été évalué dans notre étude.

La figure 4.11 nous montre la consommation de chaque Led. On constate qu'indépendamment du type de Led, le courant absorbé varie de 1.4 mA à 4.4 mA. Par rapport à la consommation d'énergie du processeur, l'utilisation des leds est à éviter dans l'objectif des protocoles de communication basse consommation. En ce qui concerne le type de diodes électroluminescentes : différentes résistances en série sont utilisées pour protéger celles-ci, les tensions de seuil sont différentes suivant la couleur des LEDs et les courants consommés dépendent aussi de la couleur des LEDs (des types de semi-conducteurs). Ainsi les caractéristiques de consommation sont fonction de la couleur des LEDs.

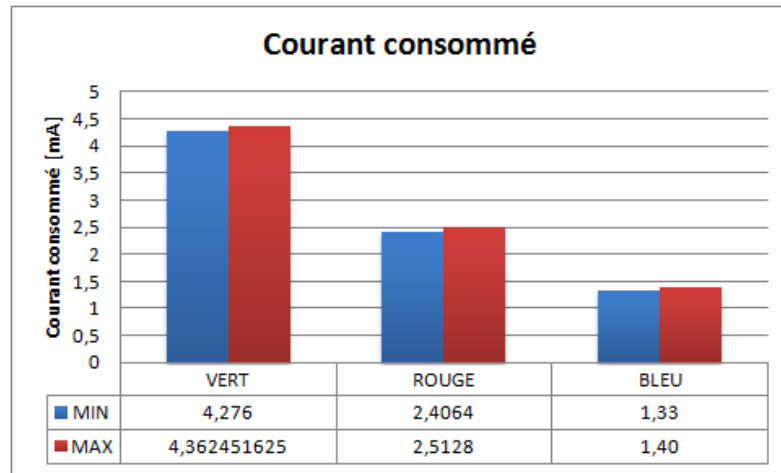


FIGURE 4.11 – Consommation des LEDs

4.4.4 Le module radio

Pour le module radio, la consommation d'énergie est en majeure partie due à l'émission, l'écoute et la réception. Toutefois, le module CC2420 possède un mécanisme de puissance d'émission configurable pour réduire sa consommation.

4.4.4.1 Phase d'émission

L'objectif de ces mesures est de caractériser la phase d'émission en variant la puissance d'émission. Le noeud envoie périodiquement un paquet de taille fixe par seconde, et après 50 envois, il ajuste sa puissance d'émission sur la prochaine puissance programmée. La période est générée par le Timer A, et juste après envoi, le noeud éteint le transceiver.

La figure 4.12 nous montre les caractéristiques mesurées dans notre étude qui constituent cette phase d'émission, avec une fréquence d'échantillonnage de $F = 50$ kHz, une puissance fixe d'émission (0 dBm) et une taille de paquet de $L = 50$ bytes.

A la lumière de la figure 4.12 obtenue par une série de mesures, nous constatons que la phase d'émission est composée de phases élémentaires, et d'une phase d'activation sous contrôle du Timer A et du CPU (la partie (0) de la figure 4.12).

La partie (1) est une phase de préparation d'envoi. Cette phase est caractérisée par le temps de démarrage du module radio et le temps de transfert de données entre la mémoire interne et le buffer de la Radio (FIFO buffer). La figure 4.13 nous montre que cette partie (1) dépend de la taille du paquet à transmettre.

Tandis que la partie (2) est équivalente à la consommation pendant la phase d'écoute. Elle dépend évidemment des paramètres de la couche MAC actuelle (CSMA), elle est donc caractérisée par le temps estimé pour accéder au canal (canal libre).

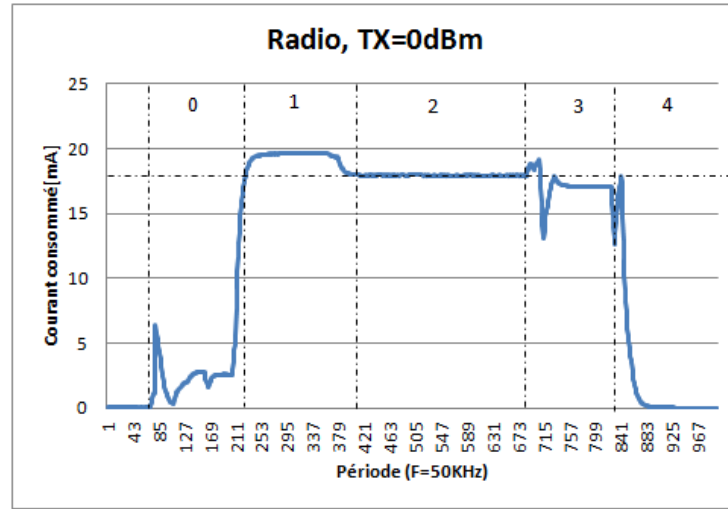
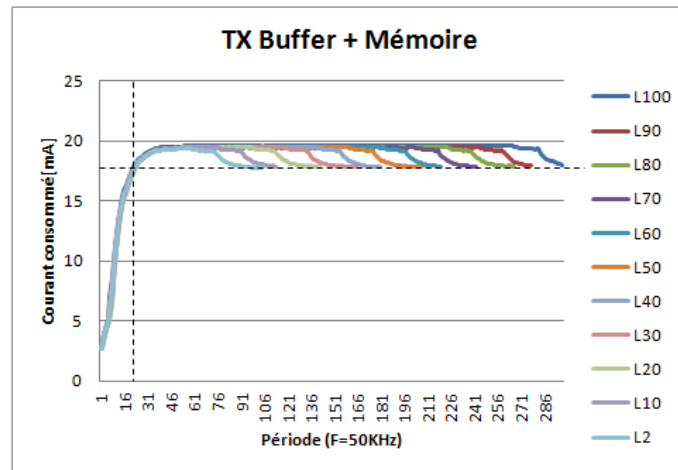
FIGURE 4.12 – Phase d’émission mesurée, $L=50$ bytes, 0 dBm

FIGURE 4.13 – Transfert mesuré entre le Buffer et la mémoire, avec une taille du paquet variable

La partie (3) correspond, à la phase d’émission du paquet proprement dit. L’émission est caractérisée par le temps de basculement vers le mode d’émission et le temps d’émission. Les mesures de la figure 4.14 nous démontre que le temps d’émission est proportionnelle à la taille du paquet. D’ailleurs, le niveau de consommation dépend de la configuration du niveau de puissance d’émission actuelle. La figure 4.15 montre les mesures de ces différentes consommations avec une tension d’alimentation de 3V et une taille de paquet de $L = 50$ bytes. Dans cette figure, on observe directement une réduction de 50% entre les puissances de transmission maximale et minimale.

108 Chapitre 4. Caractérisation des consommations d'énergie d'un noeud

De ce fait, nous entrevoyons que le contrôle de puissance s'avère être une source d'économie d'énergie intéressante pour notre objectif.

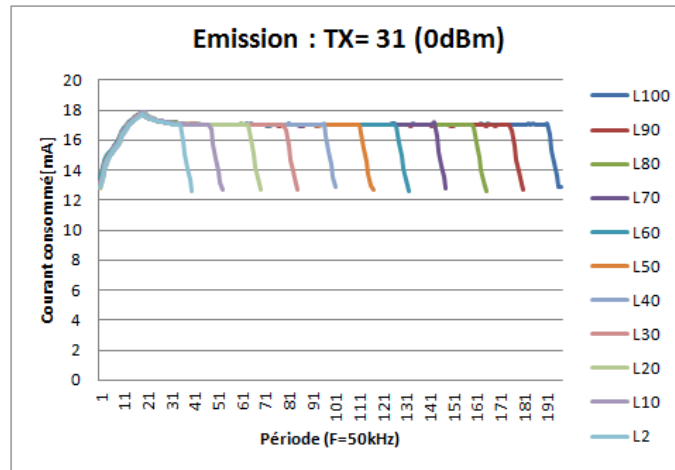


FIGURE 4.14 – Emission à TX=0dBm avec la taille du paquet variable

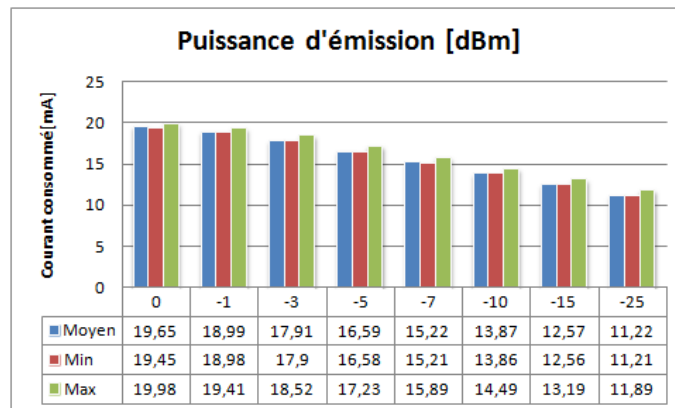


FIGURE 4.15 – Comparaison du courant consommé de différentes puissances d'émissions

Le tableau 4.3 récapitule ces résultats tout en comparant les valeurs mentionnées dans la documentation du Telosb. Pour la puissance d'émission de *0dBm*, nos résultats confirment les limites inférieures et supérieures de la documentation. D'ailleurs, à l'aide de nos résultats de mesure, nous avons pu compléter les valeurs des autres puissances du tableau.

Radio mode		Mesures: V=3V – Clock= 4Mhz			DATASHEET TELOS6			
		Radio + CPU			Only Radio	Radio + CPU		Only Radio
		Moy	Min	Max		Max	Min	
Reception		19,85	19,74	19,95	18,44	23	21,8	19,7
Emission	TX 31	19,65	19,45	19,98	17,54	21	19,5	17,4
	TX 27	18,99	18,98	19,41	16,71	-	-	16,5
	TX 23	17,91	17,9	18,52	15,52	-	-	15,2
	TX 19	16,59	16,58	17,23	13,98	-	-	13,9
	TX 15	15,22	15,21	15,89	12,55	-	-	12,5
	TX 11	13,87	13,86	14,49	11,15	-	-	11,2
	TX 7	12,57	12,56	13,19	10,01	-	-	9,9
	TX 3	11,22	11,21	11,89	8,55	-	-	8,5
Ecoute		-	-	-	17,85	-	-	-

TABLE 4.3: Comparaison des courants consommés du module radio en mode émission et réception.

4.4.4.2 Phase de réception

Dans cette sous-section, nous nous intéressons à la consommation pendant la phase de réception. Pour ce faire, nous avons synchronisé la période de réveil de l'émetteur et du récepteur en utilisant le Timer fonctionnant périodiquement. La distance qui sépare les deux noeuds est de 4 m. L'émetteur est configuré avec une puissance d'émission maximale de 0 dBm. L'objectif est de caractériser le processus de réception en faisant varier la taille du paquet.

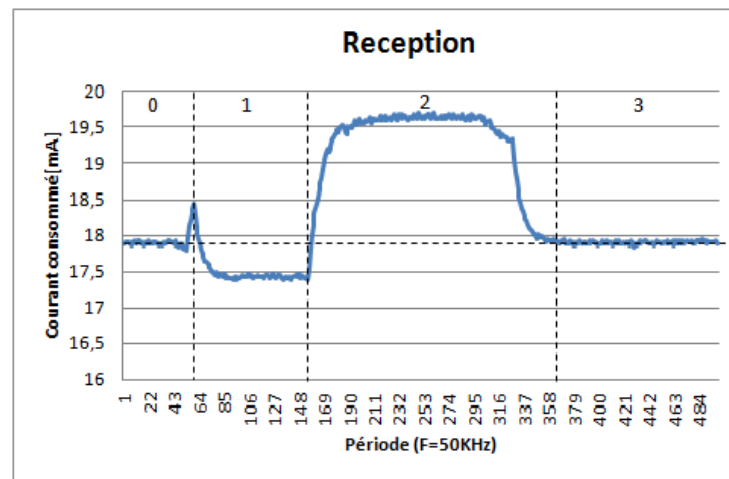


FIGURE 4.16 – Phase de reception, L=50 bytes, distance RxTx= 4 m, Puissance(Tx)=0 dBm.

A l'aide de la figure 4.16, la phase de réception peut être décomposée en quatre parties distinctes. La partie (0) correspond à la phase d'écoute. Elle est caractérisée par le temps de propagation ainsi que le temps d'accès au canal de l'émetteur (sous-

couche MAC). Ensuite, la partie (1) constitue la phase de réception proprement dite. Cette phase est caractérisée par le temps de basculement vers le mode de réception et le temps de réception. La figure 4.17 nous présente la phase de réception en fonction de la taille du paquet (les phases sont identiques). Puis, la partie (2) est caractérisée par le temps de transfert de données entre le buffer du Radio et la mémoire interne. La figure 4.18 nous montre que cette partie est fonction de la taille du paquet. Enfin, la partie (3) est équivalente à la consommation pendant la phase d'écoute.

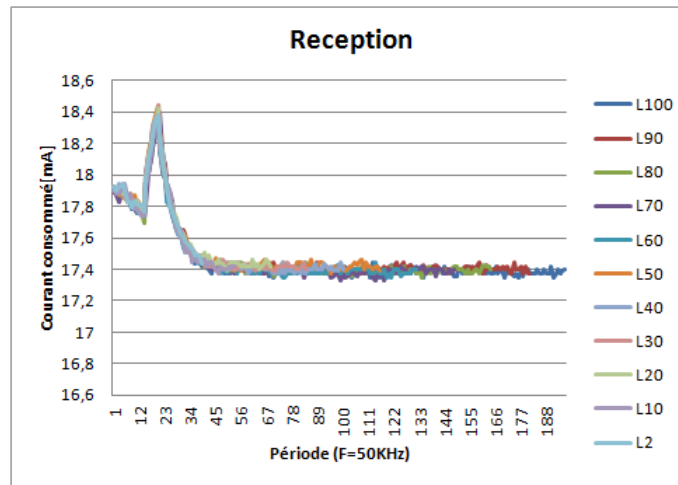


FIGURE 4.17 – Mode Reception avec Taille du paquet variable

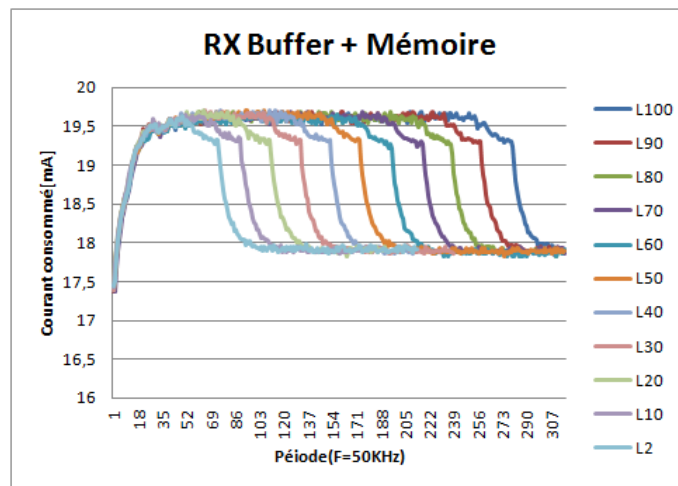


FIGURE 4.18 – Transfert entre le buffer RX et la mémoire

Le tableau de résumé 4.3 nous permet d'en conclure que la phase d'écoute consomme plus d'énergie que l'émission et la reception.

4.4.5 La mémoire flash externe

Telos utilise une mémoire flash ST-M25P80 fonctionnant à 40 MHz pour le stockage de données externes et des codes [139]. La mémoire flash contient 1024Ko de données et est décomposée en 16 segments, ayant chacun une taille de 64Ko. Elle partage les bus de communication SPI avec l'Emetteur-Récepteur CC2420. Nous avons aussi caractérisé cet élément en terme de courant consommé.

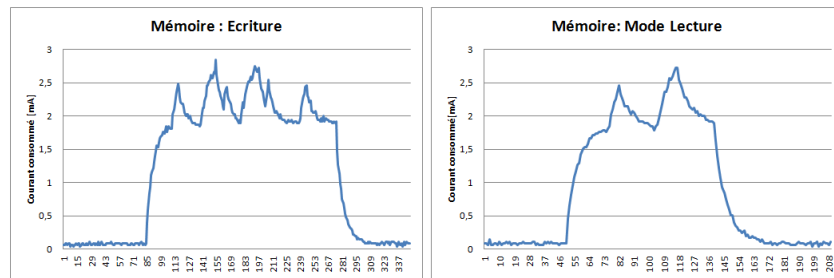


FIGURE 4.19 – Flash : courant consommé par la mémoire ST-M25P80 - 1 byte.

L'objectif des mesures de cette étape est d'analyser le courant consommé par deux opérations courantes de la mémoire flash (écriture et de lecture). La figure 4.19 résume les résultats des mesures effectuées pour les deux opérations : (a) Ecriture et (b) Lecture. Nous constatons que le besoin en courant pour une lecture est le même que celui pour l'écriture, soit de 2.5mA. En revanche le temps d'écriture est d'environ deux fois plus long que le temps de lecture. Ainsi, pour une même longueur de données (1 ou 40 bytes), l'écriture requiert logiquement beaucoup plus d'énergie que la lecture. L'énergie requise par la mémoire flash présente un effet non négligeable dans la consommation globale du noeud (2.6 mA sous 3 V au maximum).

4.5 Modélisation et synthèse

L'objectif des mesures présentées précédemment est non seulement de déterminer les différentes consommations de chaque module d'un noeud, mais aussi d'extraire les éléments pertinents concernant la consommation énergétique de chaque composant fonctionnel. Ceci afin de construire un modèle de consommation énergétique générique qui prend en compte les particularités associées aux différents paramètres des modules classiques des composants présents (débit binaire, fréquence de bus ...), tout en restant à un niveau de modélisation analytique acceptable.

Nos résultats de mesures ont montré que la variation de la consommation d'énergie entre les modules évalués est significative. En effet, nous pouvons en déduire que l'énergie consommée par un noeud de capteur peut être attribuée à sept principales sources de consommation d'énergie de base :

- de l'état du microcontrôleur,
- de l'émission et de la réception,
- de l'énergie transitoire : transitions on-off, off-on, off-sleep, sleep-off, LPMx-LPMy,
- du Timer,
- enfin du transfert et du stockage de données.

La consommation totale d'énergie d'un noeud sera obtenue par la somme de la consommation d'énergie de chacun des composants du noeud.

4.5.1 Le microcontrôleur

En ce qui concerne le microcontrôleur MSP430, les différentes consommations dépendent de ses cinq modes de faibles puissances possibles. Nous avons constaté que les deux modes de puissances (LPM0-LPM1) requièrent une consommation d'énergie considérable. Alors que les autres modes aboutissent à la réduction significative de la consommation.

Ainsi le noeud doit rester sur un mode donné lorsque le processeur ne fait pas d'opérations ou n'est pas piloté par d'autres périphériques. Seront pris en compte la consommation du mode et la durée de celui-ci. Notons par ailleurs, que l'opération actuelle ou l'algorithme exécuté pendant un mode donné n'affecte pas le courant consommé mesuré.

Ainsi, l'énergie consommée peut être calculée en multipliant la puissance du mode actuel par la période de temps pendant laquelle le processeur est resté actif. En revanche pour les transitions inter-modes, le délai de retour pour la transition d'un mode à un autre ou vers l'état actif complet doit être considéré. Ainsi, finalement, la consommation d'énergie du processeur E_{mcu} peut être exprimée par :

$$\begin{aligned} E_{mcu}^{(i,k)} &= E_{mcu-mode} + E_{trans-mode} \\ &= \sum_{i=1}^n P_{mcu}^{(i)} \times T_{mcu}^{(i)} + \sum_{k=1, i \neq k}^m P_{mcu}^{(i,k)} \times T_{mcu}^{(i,k)} \end{aligned} \quad (4.5)$$

Où : $P_{mcu}^{(i)}$ est la puissance consommée en mode de puissance (i) , $T_{mcu}^{(i)}$ est le temps passé dans ce mode. $P_{mcu}^{(i,k)}$ et $T_{mcu}^{(i,k)}$ représentent la puissance et le temps pour les transitions entre deux modes $(i \text{ et } k)$. m et n sont respectivement le nombre d'état et la transition vers un autre mode. Pour une raison de simplicité, on suppose que cette transition est linéaire. Par conséquent, l'énergie consommée pendant la transition

110 Chapitre 4. Caractérisation des consommations d'énergie d'un noeud

nées par l'intermédiaire de bus SPI. En réalité, pour s'activer depuis l'état de veille, l'architecture radio passe par ces différentes étapes (Fig. 4.20) :

- démarrage de l'oscillateur externe du module radio,
- Polarisation des étages HF de la radio,
- Démarrage du synthétiseur de fréquence (PLL),
- Passage en mode émission ou réception.

Chaque étape prend un temps non négligeable et donc une consommation non négligeable à chaque réveil. Finalement, on peut, d'une manière générale, diviser l'énergie totale consommée en mode émission E_{Emis} en cinq composantes principales (fig.4.12) :

$$E_{Emis} = E_{cmd} + E_{dem} + E_{mcu-tx} + E_{cca} + E_{tx} \quad (4.7)$$

Où :

E_{cmd} est l'énergie consommée liée à la commande initiée par le Timer et le passage du microcontrôleur de LPM3 à LPMA. Dans la pratique, cette consommation est constante, et peut être exprimée par :

$$E_{cmd} = E_{timer} + E_{mcu}^{(3,A)} \quad (4.8)$$

E_{dem} est l'énergie consommée pendant le démarrage de la radio. En principe, c'est l'énergie consommée pendant la transition de l'état en veille vers l'état actif du module radio, selon la (fig.4.12), cette énergie peut être exprimée par :

$$E_{dem} = \frac{(P_{mcu}^{(A)} + P_{mcu+radio}) \times T_{radio}^{on}}{2} \quad (4.9)$$

Du point de vue conceptuel, nous pouvons considérer que les modules radio et microcontrôleur démarrent l'un après l'autre. Par conséquent, on peut exprimer, de façon générale, l'énergie de transition d'un état à l'autre, $E_{radio}^{(i,k)}$, du module radio par :

$$\begin{aligned} E_{radio}^{(i,k)} &= \sum_{k=1, i \neq k}^m P_{radio}^{(i,k)} \times T_{radio}^{(i,k)} \\ &= \sum_{k=1, i \neq k}^m \frac{(P_{radio}^{(i)} + P_{radio}^{(k)}) \times T_{radio}^{(i,k)}}{2} \end{aligned} \quad (4.10)$$

Où : $P_{radio}^{(i)}$ est la puissance consommée pendant l'état (i) , $T_{radio}^{(i)}$ est le temps passé dans cet état. $P_{radio}^{(i,k)}$ et $T_{radio}^{(i,k)}$ sont la puissance et le temps pour les transitions entre deux états $(i$ et $k)$.

L'énergie E_{mcu-tx} correspond à la consommation pendant le transfert de données. Cette énergie est équivalente à l'énergie utilisée pour la lecture de données dans la mémoire interne et de l'écriture dans la TXFIFO [2, p.31]. Sachant que l'émission

est assurée par le bus SPI, la vitesse d'émission (écriture) est conditionnée par la fréquence d'écriture du TXFIFO. Dans la pratique, cette fréquence doit être inférieure ou égale au débit symbole du modulateur. En revanche, la lecture en principe ne dépend que la fréquence du bus de communication. Par conséquent, nous pouvons supposer que la fréquence de lecture est égale au débit symbole du modulateur. L'énergie E_{mcu-tx} pour transférer un paquet de L bits dépend alors de la longueur du paquet et du temps de transfert. Soit plus précisément du débit symbole D_s en kbps ainsi que la largeur de bus l en bits et bien entendu de la puissance totale consommée pendant la transfert (mcu+radio, Table 4.3) :

$$\begin{aligned} E_{mcu-tx} &= P_{mcu+tx} \times T_{mcu+tx} \\ &= P_{mcu+tx} \times \frac{1+l}{l \times D_s} \cdot L \end{aligned} \quad (4.11)$$

Ensuite, après avoir transféré les données, le module radio bascule en mode écoute. Dans cet état, de nombreuses parties du circuit radio sont actifs, et d'autres peuvent être éteintes. La plupart des modules radio en mode écoute, a une consommation d'énergie presque égale à l'énergie consommée en mode de réception. Cela n'est pas vrai pour le CC2420. Cela peut s'expliquer par le fait que, le microcontrôleur bascule seulement en mode LPM1 veille afin de réagir rapidement pour initier la transmission. La consommation d'énergie en mode écoute E_{cca} dépend alors du temps estimé pour que le canal soit libre T_{cca} et bien entendu de la puissance consommée, P_{cca} , équivalent à l'écoute (Table.4.3) :

$$E_{cca} = P_{cca} \times T_{cca} \quad (4.12)$$

Dans l'équation des cinq composantes principales, E_{tx} est l'énergie consommée pendant l'émission proprement dite. On sait que le signal d'émission provient du synthétiseur de fréquence, après la conversion des données numériques par le DAC. Le signal analogique filtré est transposé à la fréquence d'émission via un mélangeur de fréquence puis amplifié par l'amplificateur de puissance (PA). En effet, dans le cas d'émission d'un paquet de L bits et d'un débit symbole D_s , on admet alors pour l'émission, la consommation énergétique suivante :

$$\begin{aligned} E_{tx} &= P_{tx} \times T_{tx} \\ &= P_{tx} \times \frac{L}{D_s} \end{aligned} \quad (4.13)$$

P_{tx} correspond à la somme de puissance consommée par le mélangeur de fréquence, le synthétiseur de fréquence, les différents filtres de la chaîne de communication numérique et enfin par l'amplificateur de puissance. Toutefois, la puissance d'émission est paramétrable, les valeurs mesurées sont mentionnées dans le tableau (4.3) en fonction de l'indice de puissance.

4.5.2.2 Réception

En réception, le principe reste le même qu'en émission. En effet, le réveil d'un noeud est initié également par le Timer (fig.4.12) et le temps de réveil est souvent synchronisé à celle de l'émetteur. Ensuite, bien évidemment, le module passe en mode écoute après le démarrage, avant de basculer en mode réception. Donc, la puissance consommée au démarrage est liée au réveil du récepteur. Elle est égale, comme dans le cas d'un émetteur, à $E_{cmd} + E_{dem}$. A la différence, le processus est inversé en réception : dès l'arrivée du signal, un amplificateur faible bruit (LNA) vient amplifier le signal reçu. Ensuite, un mélangeur vient transposer le signal RF à une fréquence intermédiaire très faible. L'inconvénient de ce mécanisme de transposition de fréquence concerne l'ajout d'un signal non désirable à la fréquence image de la fréquence de transposition. Ce qui justifie l'ajout en fin de chaîne d'un filtre de réjection de la fréquence image avec un amplificateur de fréquence intermédiaire. L'énergie dépensée pour le décodage dépendra quant à elle des codes correcteurs implémentés au préalable. La prise en compte de toutes ces puissances mises en jeu par le processus de réception nous permet de définir l'énergie totale consommée E_{Recv} pendant cette phase par la relation (4.14) - des termes sont déjà connus :

$$E_{Recv} = E_{cmd} + E_{dem} + E_{cca} + E_{rx} + E_{mcu-rx} \quad (4.14)$$

Les figures (4.12) et (4.16) nous démontrent que l'énergie liée à l'écoute E_{cca} est identique pour le récepteur et l'émetteur, sa valeur est mentionnée dans le tableau (4.3).

E_{rx} est l'énergie consommée pendant la réception du paquet proprement dit. Dans la pratique, le signal de transmission provient également du même synthétiseur de fréquence. De ce fait, le temps nécessaire pour recevoir un paquet de L bit dépend également du débit symbole D_s du démodulateur. Cette énergie peut donc être exprimée par :

$$\begin{aligned} E_{rx} &= P_{rx} \times T_{rx} \\ &= P_{rx} \times \frac{L}{D_s} \end{aligned} \quad (4.15)$$

où : P_{rx} correspond à la somme des puissances consommées mises en jeu par le processus de réception. Sa valeur est également mentionnée dans le tableau (4.3).

L'énergie E_{mcu-rx} correspond à la consommation durant le transfert de données de RXFIFO vers la mémoire interne du microcontrôleur. Cette énergie dépend alors du temps nécessaire pour la lecture de données reçues dans le mémoire tampon RXFIFO et l'écriture de données dans la mémoire interne du microcontrôleur. Rappelons que le temps de l'écriture est obtenu à l'aide de la fréquence des opérations d'effacement et d'écriture. De plus, la transmission est également assurée par

le bus SPI, donc la vitesse de transmission est influencée par la fréquence d'écriture d'un bit de la mémoire interne du microcontrôleur. Toutefois, la vitesse de transmission doit être inférieure au débit symbole du démodulateur [2, p.31]. Nous pouvons donc supposer que la fréquence de transmission soit égale au débit symbole théorique [2, p.31]. E_{mcu-rx} peut être déterminée par la relation (4.16) :

$$\begin{aligned} E_{mcu-rx} &= P_{mcu+rx} \times T_{mcu+rx} \\ &= P_{mcu+rx} \times \left(\frac{1+l}{l \cdot D_s} \times L + T_{eff} \right) \end{aligned} \quad (4.16)$$

où T_{eff} : temps d'effacement de données.

Il est à noter qu'à la fin de la phase de transmission (que ce soit en réception ou en émission), le noeud doit éteindre ou mettre en veille son module radio pour conserver son énergie. Cependant, le coût énergétique de transition d'un état à un autre n'est pas toujours négligeable. Pour les déterminer, on considère l'équation (4.10).

Finalement, la consommation d'énergie d'un noeud durant une communication peut être déterminée par la relation (4.17) :

$$\begin{aligned} E_{Noeud} &= E_{cmd} + E_{mcu} + E_{radio} + E_{mcu+radio} \\ &= \underbrace{E_{timer} + \frac{(P_{mcu}^{(A)} + P_{mcu+radio}) \times T_{radio}^{on}}{2}}_{\text{Demarrage du noeud}} \\ &\quad + \underbrace{\sum_{i=1}^n P_{mcu}^{(i)} \times T_{mcu}^{(i)} + \sum_{k=1, i \neq k}^m P_{mcu}^{(i,k)} \times T_{mcu}^{(i,k)}}_{\text{mode + transition du mcu}} \\ &\quad + \underbrace{P_{tx} \times \frac{L}{D_s}}_{\text{Emission}} + \underbrace{P_{rx} \times \frac{L}{D_s}}_{\text{Reception}} + \underbrace{P_{cca} \times (T_{cca}^{tx} + T_{cca}^{rx})}_{\text{Ecoute}} + \underbrace{\sum_{k=1, i \neq k}^m P_{radio}^{(i,k)} \times T_{radio}^{(i,k)}}_{\text{transition radio}} \\ &\quad + \underbrace{P_{mcu+tx} \times \frac{1+l}{l \times D_s} \times L}_{\text{Transfert mcu txfifo}} + \underbrace{P_{mcu+rx} \times \left(\frac{1+l}{l \cdot D_s} \times L + T_{eff} \right)}_{\text{Transfert rxfifo mcu}} \end{aligned} \quad (4.17)$$

4.5.3 Synthèse des énergies consommées

Pour pouvoir analyser les résultats des mesures, nous avons présenté les résultats et les modèles sous forme graphique regroupant la phase d'émission, le transfert de données entre la mémoire interne du microcontrôleur et la mémoire tampon d'émission (MCU-TXFIFO), la phase de réception et enfin le transfert de données entre la mémoire tampon de réception et la mémoire interne du microcontrôleur (RXFIFO-MCU). Chaque courbe représente les résultats des mesures et les modèles

adoptés. Les autres consommations sont considérées constantes, notamment le mode de puissance du microcontrôleur, les différentes transitions, les leds, le Timer. Tandis que la phase constituant la transmission (réception ou émission) est représentée en fonction de la taille de trames ainsi que la fréquence de fonctionnement (Débit symbole). Pour l'émission de paquets, les résultats se trouvent dans la figure 4.21. Dans cette figure, la courbe fait référence à un débit de transmission $D_s = 250\text{kbps}$.

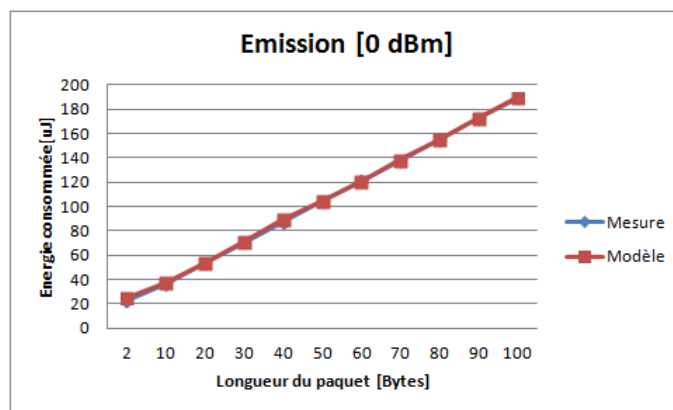


FIGURE 4.21 – Emission : comparaison modèle et mesures

Nous pouvons remarquer que le modèle et l'expérience de consommation énergétique croissent de façon linéaire et se superposent presque parfaitement. De plus, lorsque la taille du paquet de données augmente, la consommation d'énergie augmente aussi. Ceci s'explique essentiellement par le fait que la durée d'émission d'un paquet dépend uniquement du débit de transmission. La figure 4.22 montre l'énergie consommée pendant le transfert du paquet MCU-TXFIFO, toujours suivant la taille du paquet. On constate également que la consommation énergétique évolue linéairement en fonction de la taille du paquet. Cependant, les écarts que nous pouvons constater entre le modèle et la mesure entre le paquet 2 – 50 octets sont respectivement de l'ordre de 6.28% et 1.34%. Ce qui reste convenable.

Du point de vue de la réception, nous avons représenté dans la figure 4.23, l'évolution du coût énergétique de la réception d'un paquet avec un débit de transmission $D_s = 250\text{kbps}$. Cette figure montre également que la consommation en énergie varie de façon linéaire en fonction de la taille du paquet. Ceci s'explique tout simplement par le fait que la consommation d'énergie pour la réception dépend du débit de transmission et de la taille du paquet envoyé.

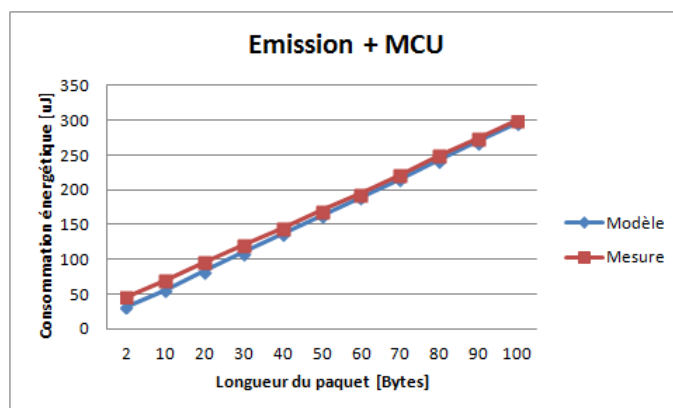


FIGURE 4.22 – Transfert de données MCU-TXFIFO

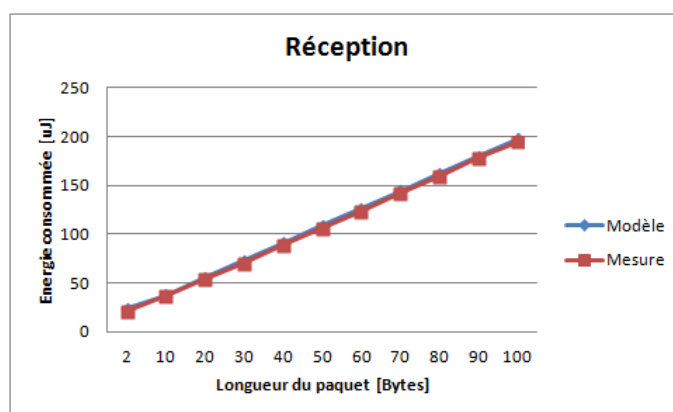


FIGURE 4.23 – Réception : comparaison modèle et mesures

La figure 4.24 représente l'énergie consommée pendant le transfert du paquet RXFIFO-MCU en fonction de la taille du paquet. On constate également que la consommation énergétique évolue linéairement en fonction de la taille du paquet. Cependant, nous pouvons remarquer un léger écart entre le modèle et la mesure pour des paquets de 2 à 50 octets. Ces écarts sont respectivement de l'ordre de 5.78% et 0.74%. Cela peut être dû au mécanisme du protocole d'échange interne du bus SPI.

Pour finir, afin d'estimer l'influence de chaque module sur la consommation globale du nœud, la figure 4.25 nous présente les différentes consommations de chaque phase de transmission (émission et réception). À la lumière de cette figure, on peut observer que le module radio CC2420 consomme beaucoup plus d'énergie que le microcontrôleur. Cela peut s'expliquer par le fait que ce dernier est toujours à un état de puissance par défaut c'est-à-dire en mode LPM3 (pour conserver l'énergie). Sauf

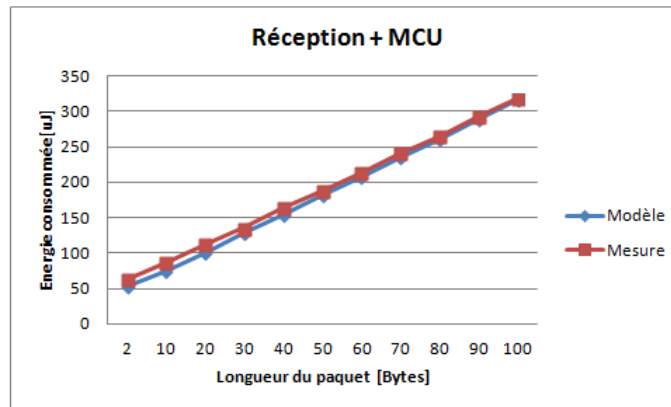


FIGURE 4.24 – Transfert de données RXFIFO-MCU : comparaison modèle et mesures

au démarrage du noeud avant de processus de transmission. Et l'énergie consommée par le Timer pour déclencher le démarrage est d'environ 6 fois moins que celle du C2420 (vis à vis des valeurs mesurées). Tandis que la consommation du microcontrôleur est d'environ 5.5 fois moins que celle du module radio lorsqu'il est dans l'état actif. Ainsi, en étudiant séparément les différents modules (mcu, radio), c'est la phase de transmission de données (émission ou réception) qui est la plus consommatrice en énergie [108]. En analysant les transferts de données MCU-TXFIFO et RXFIFO-MCU, on note qu'ils présentent un coût énergétique très important tant en émission qu'en réception. Par rapport à l'émission et la réception, les consommations associées tendent respectivement vers 61% et 67%.

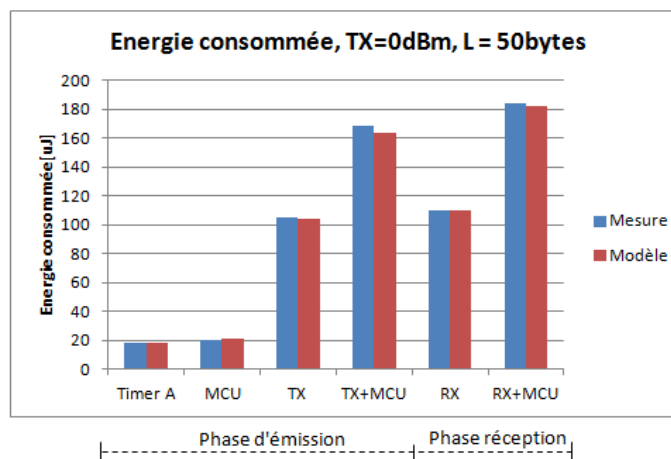


FIGURE 4.25 – Comparaison des consommations énergétiques - CC2420(TX-RX) et MSP430(MCU).

4.6 Conclusions

Nous avons consacré ce chapitre à la caractérisation des consommations énergétiques d'un noeud : le TelosB. L'objectif a été de caractériser les modes de fonctionnement et de mesurer l'énergie consommée par la plateforme matérielle utilisée. De ces résultats expérimentaux, nous avons pu identifier et quantifier la consommation énergétique de chaque module fonctionnel. Nous avons pris en compte les différents états observés dans un fonctionnement de base d'un noeud. Ainsi nous avons effectué nos mesures depuis le démarrage du noeud jusqu'à l'émission ou la réception des données. Cela nous a permis d'obtenir un modèle énergétique fiable, précis et valide pour notre plateforme matérielle. Nos résultats confirment que la quantité d'énergie dissipée en émission est liée à la taille du paquet de façon linéaire et fonction de la puissance d'émission. Tandis que la réception ne dépend seulement que de la taille du paquet pour une puissance de réception donnée.

Par ailleurs, on a aussi vu que la phase de transfert de données entre le microcontrôleur et le transceiver est consommatrice en termes d'énergie. Cela est influencé par l'échange de données entre deux mémoires internes. La dissipation d'énergie dépend du temps de lecture et d'écriture de deux mémoires. Et elle est surtout liée à la capacité et aux caractéristiques (largeur) du bus de communication qui assure ce transfert de données. Ce point intéressant a été intégré à un modèle associé aux différentes autres phases et mode de fonctionnement du noeud.

Ce travail nous a aussi permis de voir les améliorations futures du thème de ce chapitre. En effet dans le choix qui a été fait des fonctions embarquées classiques qui ont été caractérisées, nous voyons en fin d'étude :

- que des phénomènes physiques associés aux capteurs, leur polarisation, leur conditionnement et leur temps de réponse, vont augmenter notablement la consommation énergétique. Pour exemple le capteur d'humidité SHT1x, embarqué sur le TelosB à un temps de réponse typique à 63% de 8 secondes. Ce qui ramener à la fréquence d'horloge et les time-slots d'émission des noeuds est important. De même pour se placer en état de veille le capteur a besoin de 11 ms. De nombreux capteurs ont une contrainte d'auto-échauffement qui peut fausser la mesure et la chaîne de commande. Des contraintes telles qu'une interrogation préconisée toutes les 1 voire 2 secondes, peuvent alourdir encore le bilan énergétique. Enfin certains capteurs demandent un pré-conditionnement, une polarisation ou un préchauffage (capteur de CO2 par exemple). Ces éléments viennent allonger la liste des consommations énergétiques.
- Il nous semble donc important d'envisager un terme : $\alpha.E_{\text{CapteursAnalogiques}} + \beta.E_{\text{CapteursNumeriques}}$ tenant compte des remarques précédentes et du temps d'utilisation des capteurs considérés.
- une modélisation sous forme de machine d'état du noeud complet serait une

avancée vers l'évaluation locale de la consommation. Evaluation qu'il faudrait intégrée aux trames d'informations pour un autre travail d'optimisation de la durée de vie du réseau.

Nous pensons que ces observations sont généralisables à d'autres types de noeud des RCSFs (autres que TelosB). Et la question de l'impact de l'utilisation de ce nouveau modèle sur les optimisations présentées dans la littérature, est un point très intéressant.

Conclusion générale et perspectives

Sommaire

5.1 Conclusions générales	123
5.1.1 L'algorithme de clustérisation centralisé	124
5.1.2 L'algorithme de clustérisation distribué	125
5.1.3 L'algorithme décentralisé	125
5.1.4 L'algorithme multihop des protocoles de routage	126
5.1.5 Caractérisation des consommations énergétiques d'un noeud .	126
5.2 Perspectives	127
5.2.1 Aspect protocolaire	127
5.2.2 Aspect expérimental	128

5.1 Conclusions générales

L'utilisation des RCSFs ne cesse de s'accroître ces dernières années en raison des opportunités en matière d'application dans les différents domaines tels que : le militaire, la santé, l'environnement, la sécurité publique, etc. L'un des défis majeurs de ces réseaux est la contrainte énergétique des noeuds. Cela demande une gestion plus stricte des ressources disponibles lors des conceptions des protocoles de communication et à différentes couches de la pile protocolaire afin de prolonger la durée de vie du réseau. En considérant la couche réseau, l'efficacité énergétique est liée à l'organisation hiérarchique des noeuds ainsi qu'au protocole de communication. L'analyse approfondie de différentes solutions de la littérature a guidé nos travaux vers le concept des protocoles de communication basés sur le clustering pour répondre à nos objectifs.

Dans cette optique, nous avons porté un grand intérêt à l'étude des différents algorithmes construction des clusters, pour proposer un algorithme de routage adapté au problème de scalabilité et de passage à l'échelle. Tout au long de ce processus

d'amélioration et d'optimisation, l'accent a été mis sur la notion d'équilibrage de la consommation énergétique des noeuds de sorte à maximiser la durée de vie du réseau. Dans un premier temps, nous avons optimisé un protocole centralisé à un saut (one-hop) avec l'objectif de mieux équilibrer la charge, et d'améliorer la stabilité de la topologie des clusters pour garantir la couverture de toutes les zones. Ensuite, dans l'objectif de permettre le fonctionnement avec un grand nombre de noeuds, nous avons optimisé un algorithme distribué. L'étude comparative des résultats obtenus de ces deux algorithmes nous a orientés vers la décentralisation de l'algorithme centralisé pour mettre en évidence l'effet du critère de choix du cluster-head et l'équilibrage énergétique qui régissent le fonctionnement de l'algorithme one-hop. Enfin, La maîtrise du concept des protocoles clustering et l'étude approfondie des protocoles de routage existants nous ont permis alors de proposer un nouveau protocole en introduisant un routage multisaut dans la structure de clusters pour résoudre le problème de passage à l'échelle.

5.1.1 L'algorithme de clustérisation centralisé

Dans cette thèse, nous avons proposé, deux protocoles centralisés dénotés : k-LEACH et sa version optimisée k-LEACH-VAR. L'objectif de ces protocoles est d'optimiser la durée de vie et l'efficacité énergétique du réseau en formant une bonne distribution des clusters qui couvre l'ensemble des noeuds dans le domaine étudié. Ces deux protocoles sont basés sur l'algorithme de clustérisation à un saut qui utilise l'énergie restante et la variance d'énergie consommée du noeud comme métriques de sélection des cluster-heads. L'idée de base est de sélectionner les cluster-heads parmi les noeuds ayant l'énergie restante la plus élevée et la plus faible variation de consommation d'énergie.

Dans une première évaluation, nous avons analysé l'impact de la variation du rayon de couverture du cluster sur ces deux protocoles. Les résultats de simulation ont confirmé que le nombre de cluster-head diminue avec l'augmentation de rayon de couverture du cluster. Par conséquent, la durée moyenne de vie du réseau tend à augmenter lorsque le rayon de couverture du cluster augmente. En revanche, le déséquilibre énergétique persiste lorsque le nombre de cluster-head diminue. Dans une deuxième évaluation, après avoir déterminé le nombre de cluster optimal, nous avons comparé notre algorithme avec les algorithmes LEACH et LEACH-C. Le nombre de cluster optimal est un compromis du rayon de couverture de cluster qui minimise la consommation d'énergie globale du réseau mais permet de garantir la couverture du réseau entier. Les résultats des simulations montrent que notre algorithme a de meilleures performances comparé à LEACH et LEACH-C en termes de durée de vie du réseau et d'efficacité énergétique. En contrepartie, les quantités de données reçues par la station de base diminuent.

5.1.2 L'algorithme de clustérisation distribué

L'algorithme centralisé nécessite que toute information locale de tous les noeuds notamment leur réserve d'énergie pour construire leur cluster. Cela nécessite donc une communication supplémentaire entre les noeuds et la station de base. Cette communication périodique engendre l'augmentation de flux d'information. En plus, lorsque le nombre de noeuds augmente, le problème d'interférence augmente. Ce qui entraîne la perturbation du réseau. Pour pallier à ce problème, nous avons présenté un algorithme distribué appelé e-LEACH. La contribution principale d'e-LEACH est l'amélioration de la distribution des clusters tout en gardant le nombre optimal de clusters dans le réseau afin d'augmenter la durée de vie du réseau. Puis, dans la phase de sélection des cluster-heads, e-LEACH reproduit l'idée de k-LEACH en introduisant la notion d'équilibrage énergétique pour trouver une référence d'énergie restante moyenne afin de déterminer par la suite la variance de consommation énergétique.

Les résultats montrent que les données reçues offrent de meilleures performances en termes de durée de vie du réseau et de consommation énergétique. En revanche, à partir de 70% du nombre de noeuds survivants, la quantité de données reçues par la station de base diminue légèrement par rapport au protocole centralisé LEACH-C. Cette diminution est due à l'utilisation du paramètre associé à la variance de l'énergie. En effet, ce dernier ajuste progressivement le nombre de "cluster-heads" en fonction du nombre de noeud survivants.

5.1.3 L'algorithme décentralisé

Nous avons proposé un protocole k-LEACH avec une méthode décentralisée, notée K-LEACH-C2D, ayant pour objectif d'améliorer l'efficacité énergétique ainsi que l'équilibre énergétique dans l'ensemble des noeuds. La première proposition repose sur l'objectif d'équilibrage de la consommation énergétique de tous les noeuds, avec un objectif de mort simultanée de tous les noeuds. Les résultats nous montrent que l'équilibre énergétique ne peut être atteint que dans le cas où le nombre de clusters augmente. En revanche, quand le nombre de cluster diminue, le système conserve son énergie.

La deuxième proposition consiste donc à proposer un modèle de consommation globale qui décroît exponentiellement au cours du temps. Cette proposition permet d'ajuster l'évolution de la consommation globale suivant l'énergie initiale totale du réseau. Grâce à ces deux propositions, nous avons pu supprimer l'échange périodique des informations locales des noeuds avec la station de base. Ainsi, le protocole K-LEACH-C2D réduit le trafic de contrôle au sein du réseau, permettant ainsi de conserver l'énergie.

Les résultats montrent que K-LEACH-C2D offre de meilleures performances en

termes de durée de vie du réseau, la consommation d'énergie ainsi que le nombre de paquet délivré à la station de base par rapport au protocole distribué.

5.1.4 L'algorithme multihop des protocoles de routage

Afin de poursuivre cette démarche de minimisation énergétique, nous avons proposé deux protocoles de routage hiérarchique multisaut basé sur des méthodes de clustering, dénoté FRSM et CB-RSM. Ces protocoles utilisent l'algorithme de clustering probabiliste discuté précédemment notamment LEACH et e-LEACH, respectivement. Ces protocoles forment des chemins de routage de "cluster-head" en "cluster-head" afin d'effectuer la communication inter-clusters en se basant sur la construction de chemin élémentaire. FRSM utilise la puissance de la station de base pour interroger les CHs plus loin d'elle pour router leurs données. Tandis que CB-RSM utilise une technique d'auto-ordonnancement pour rendre les réseaux plus scalable.

Les résultats des simulations ont démontré que les performances de ces deux protocoles par rapport LEACH et e-LEACH. Les gains des performances de FRSM et CB-RSM augmentent d'environ 80% en termes de durée de vie, d'efficacité énergétique et de nombre du paquet délivré à la station de base.

5.1.5 Caractérisation des consommations énergétiques d'un noeud

Au vu de la littérature sur la modélisation de la consommation énergétique d'un noeud sans fil et des mesures globales sur le fonctionnement de celui-ci, nous avons voulu mieux préciser les paramètres des modèles existants. C'est pourquoi nous avons entrepris un travail expérimental sur le sujet. Ce travail s'est déroulé en plusieurs phases :

- l'identification des fonctions "embarquées" typiques et significatives permettant une approche générique de la consommation d'un noeud quelconque d'un RCSF,
- l'identification des modes de fonctionnement d'un noeud : réception, émission, émission accompagnée de transfert d'informations entre les mémoires, basculement du mode réception vers le mode émission, et les modes de fonctionnement en basse énergie,
- le développement d'une chaîne de mesure complète (à base de capteur de tension/courant, d'amplificateur d'instrumentation, de convertisseur analogique-numérique et de traitement des données) pour étudier les précédentes fonctions et modes de fonctionnement. Des mesures ont été réalisées et classées dans certaines situations en fonction des paramètres tels que la longueur de la trame de données, la puissance d'émission, et les commutations entre mode de fonc-

tionnement du noeud. Ces mesures sont comparées aux données constructrices et étendues.

- et enfin la modélisation extraite des mesures faites du noeud TelosB. Nous avons validé ce modèle global avec nos mesures et comparé avec les autres modèles de la littérature.

5.2 Perspectives

Le travail réalisé au long de cette thèse a permis d'améliorer les performances des RCSF, mais aussi d'identifier un certain nombre de problèmes. En terme de perspectives à ce travail, nous proposons pour les deux grandes orientations de ce travail les pistes suivantes :

5.2.1 Aspect protocolaire

- L'étude des différents algorithmes proposés a été faite par simulation sous Matlab. Cette étude peut être complétée par une analyse théorique après avoir explicité les formules nécessaires. L'évaluation à l'aide d'un simulateur de réseaux comme ns2 ou ns3 peut être également envisagée afin de prendre en compte l'impact de la sous-couche MAC considérée.
- Dans cette thèse, notre étude s'est focalisée sur l'algorithme de formation de cluster probabiliste, dans le contexte de noeuds statiques. Un autre aspect qui pourrait prolonger les travaux effectués, serait l'intégration d'une mobilité du noeud et/ou la station de base. En outre, il serait intéressant aussi d'intégrer et d'analyser la portabilité de notre fonction convexe dans les autres algorithmes de formation de cluster non-probabiliste, voire dans les différents protocoles : routage plat, routage géographique.
- Dans les deux phases de communication : inter-cluster ou intra-cluster sont toujours considérées comme fiables : pas de retransmission. Cependant, une retransmission est nécessaire lorsqu'un paquet transmis comporte au moins une erreur. Dans ce contexte, la fonction convexe que nous avons présentée, pourrait être toujours exploitée afin d'améliorer la qualité de service dans un environnement non idéal. Par exemple, l'intégration d'expression tenant compte l'énergie consommée par bit transmis dans le but de mettre en exergue un système permettant d'obtenir un minimum en termes de variance d'énergie par bit transmis avec succès.
- Une idée concernant une méthode de mise en veille des noeuds pourrait aussi être envisagée. En effet, dans la méthode clustering, plusieurs noeuds qui sont à proximité d'un certain événement communiquent dans le but de fusionner les

données collectées. Au lieu d'utiliser l'agrégation permettant de minimiser le nombre d'envoi des données collectées identiques par plusieurs noeuds voisins, il serait intéressant de le remplacer par un mécanisme de mise en veille systématique. Ceci permettrait d'avoir une fiabilité des informations récoltées. Ainsi, la durée de vie du réseau sera certainement prolongée en minimisant l'envoi et la réception des paquets.

- Une nouvelle approche concernant un protocole d'accès MAC pourrait aussi compléter cette étude. Sachant que, dans LEACH, après avoir élu les cluster-heads, un message d'annonce est diffusé en utilisant le protocole CSMA. De plus, il utilise également les protocoles TDMA/CDMA pour réduire les collisions dans les phases de communication de données. Cependant, ces trois protocoles sont utilisés indépendamment les uns des autres. Un seul protocole Hybride TDMA/CSMA peut être considéré afin de faciliter l'implémentation. Au cours d'un cycle TDMA, les noeuds peuvent émettre pendant un slot de temps proportionnel à la quantité de données à transmettre. De ce fait, un noeud doit être réveillé quand il est dans son slot de transmission. A ce moment, le protocole p-persistent CSMA peut intervenir pour réduire la collision entre les noeuds des autres clusters en phase de transmission et les "cluster-heads" en phase de réception dans un même slot.

5.2.2 Aspect expérimental

- Une amélioration du modèle basée sur d'autres mesures serait une extension souhaitable de ce travail. En prenant en compte par exemple l'aspect mesure des capteurs et les contraintes temporelles associées.
- Cette nouvelle modélisation devrait être une étape clé pour les futures optimisations portant sur l'énergie dans les RCSFs.

Bibliographie

- [1] *Ultra low power IEEE 802.15.4 compliant wireless sensor module, Revision B*, 2004. [Online]. Available : <http://www2.ece.ohio-state.edu/~bibyk/ee582/telosMote.pdf> (Cité en pages vii, 90, 92, 93, 95 et 98.)
- [2] (2006) Cc2420 data sheet. [Online]. Available : http://www.chipcon.com/files/CC2420_Data_Sheet_14.pdf (Cité en pages viii, 42, 95, 113, 114 et 117.)
- [3] P. Harrop and R. Das, “Wireless sensor networks (wsn) 2012-2022 : Forecasts, technologies, players,” IDTechEx, Tech. Rep., 2012. (Cité en page 7.)
- [4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks : A survey,” *Elsevier Computer Network*, pp. 393–442, March 2002. (Cité en pages 7, 8, 11 et 36.)
- [5] A. A. Abbasi and M. Younis, “A survey on clustering algorithms for wireless sensor networks,” *Elsevier Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007. (Cité en pages 7, 8, 11, 12, 13 et 36.)
- [6] M. J. Handy, M. Haase, and D. Timmermann, “Low energy adaptive clustering hierarchy with deterministic cluster-head selection,” in *Fourth IEEE Conference on Mobile and Wireless Communications Networks, Stockholm*, 2002. (Cité en pages 8, 11, 13, 14, 18, 47 et 79.)
- [7] M. Cardei, J. Wu, M. Lu, and M. O. Pervaiz, “Maximum network lifetime in wireless sensor networks with adjustable sensing ranges,” in *WiMob2005 IEEE International Conference on Wireless And Mobile Computing Networking And Communications*, vol. 3. ieee, October 2005, pp. 438–445. (Cité en page 8.)
- [8] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *Commun. ACM*, vol. 43, pp. 51–58, May 2000. (Cité en page 8.)
- [9] M. Esseghir and N. Bouabdallah, “Node density control for maximizing wireless sensor network lifetime,” *Int. J. Netw. Manag.*, vol. 18, no. 2, pp. 159–170, Mar. 2008. [Online]. Available : <http://dx.doi.org/10.1002/nem.676> (Cité en page 8.)
- [10] A. Ghosh and S. K. Das, “Coverage and connectivity issues in wireless sensor networks : A survey,” *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303 – 334, 2008. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S1574119208000187> (Cité en page 8.)

- [11] R. Mulligan and H. M. Ammari, "Coverage in wireless sensor networks : A survey," *Network Protocols and Algorithms*, vol. 2, pp. 27–53, 2010. (Cité en page 8.)
- [12] T. Voight and H. Ritter, "Solar-aware clustering in wireless sensor networks," in *Ninth IEEE Symposium on Computers and Communications*, vol. 1, 2004. (Cité en page 8.)
- [13] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini, "Modeling and optimization of a solar energy harvester system for self-powered wireless sensor networks," *Industrial Electronics, IEEE Transactions on*, vol. 55, no. 7, pp. 2759–2766, July 2008. (Cité en page 8.)
- [14] C. Alippi and C. Galperti, "An adaptive system for optimal solar energy harvesting in wireless sensor network nodes," *Circuits and Systems I : Regular Papers, IEEE Transactions on*, vol. 55, no. 6, pp. 1742–1750, July 2008. (Cité en page 8.)
- [15] P. Lee, Z. A. Eu, M. Han, and H. Tan, "Empirical modeling of a solar-powered energy harvesting wireless sensor node for time-slotted operation," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, March 2011, pp. 179–184. (Cité en page 8.)
- [16] R. Shigeta, T. Sasaki, D. M. Quan, Y. Kawahara, R. Vyas, M. Tentzeris, and T. Asami, "Ambient rf energy harvesting sensor device with capacitor-leakage-aware duty cycle control," *Sensors Journal, IEEE*, vol. 13, no. 8, pp. 2973–2983, Aug 2013. (Cité en page 8.)
- [17] P. Nintanavongsa, M. Naderi, and K. Chowdhury, "A dual-band wireless energy transfer protocol for heterogeneous sensor networks powered by rf energy harvesting," in *Computer Science and Engineering Conference (IC-SEC), 2013 International*, Sept 2013, pp. 387–392. (Cité en page 8.)
- [18] —, "Medium access control protocol design for sensors powered by wireless energy transfer," in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 150–154. (Cité en page 8.)
- [19] M. Ferrari, V. Ferrari, M. Guizzetti, D. Marioli, and A. Taroni, "Piezoelectric multifrequency energy converter for power harvesting in autonomous microsystems," *Sensors and Actuators A : Physical*, vol. 142, no. 1, pp. 329 – 335, 2008, special Issue : Eurosensors {XX} The 20th European conference on Solid-State Transducers Eurosensors 2006 Eurosensors 20th Edition. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S0924424707005481> (Cité en page 8.)

- [20] J. Gilbert and F. Balouchi, "Comparison of energy harvesting systems for wireless sensor networks," *International Journal of Automation and Computing*, vol. 5, no. 4, pp. 334–347, 2008. [Online]. Available : <http://dx.doi.org/10.1007/s11633-008-0334-2> (Cité en page 8.)
- [21] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes : Survey and implications," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 3, pp. 443–461, Third 2011. (Cité en page 8.)
- [22] J. Paradiso and T. Starner, "Energy scavenging for mobile and wireless electronics," *Pervasive Computing, IEEE*, vol. 4, no. 1, pp. 18–27, Jan 2005. (Cité en page 8.)
- [23] R. Rajesh, V. Sharma, and P. Viswanath, "Information capacity of energy harvesting sensor nodes," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, July 2011, pp. 2363–2367. (Cité en page 8.)
- [24] Y. Zhang, S. He, J. Chen, Y. Sun, and X. Shen, "Distributed sampling rate control for rechargeable sensor nodes with limited battery capacity," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 6, pp. 3096–3106, June 2013. (Cité en page 8.)
- [25] R. Min, M. Bhardwaj, S. Cho, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Low-power wireless sensor networks," in *VLSI Design, 2001. Fourteenth International Conference on*, 2001, pp. 205–210. (Cité en page 8.)
- [26] J. Polastre, R. Szewczyk, and D. Culler, "Telos : enabling ultra-low power wireless research," in *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, April 2005, pp. 364–369. (Cité en pages 8 et 91.)
- [27] B. H. Calhoun, D. C. Daly, N. Verma, D. F. Finchelstein, D. D. Wentzloff, A. Wang, S. Cho, and A. P. Chandrakasan, "Design considerations for ultra-low energy wireless microsensor nodes," *Computers, IEEE Transactions on*, vol. 54, no. 6, pp. 727–740, 2005. (Cité en page 8.)
- [28] M. Hempstead, M. J. Lyons, D. Brooks, and G.-Y. Wei, "Survey of hardware systems for wireless sensor networks," *Journal of Low Power Electronics*, vol. 4, no. 1, pp. 11–20, 2008. [Online]. Available : <http://www.ingentaconnect.com/content/asp/jolpe/2008/00000004/00000001/art00002> (Cité en page 8.)
- [29] Y. Lee, Y. Kim, D. Yoon, D. Blaauw, and D. Sylvester, "Circuit and system design guidelines for ultra-low power sensor nodes," in *Proceedings of the 49th Annual Design Automation Conference*, ser. DAC '12. New

- York, NY, USA : ACM, 2012, pp. 1037–1042. [Online]. Available : <http://doi.acm.org/10.1145/2228360.2228548> (Cit  en page 8.)
- [30] “Ieee standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks specific requirements part 15.4 : Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans),” *IEEE Std 802.15.4(2003)*, pp. 01–670, 2003. (Cit  en pages 8 et 96.)
- [31] Z. Xiong, A. D. Liveris, and S. Cheng, “Distributed source coding for sensor networks,” *Signal Processing Magazine, IEEE*, vol. 21, no. 5, pp. 80–94, 2004. (Cit  en page 8.)
- [32] S. Cui, A. J. Goldsmith, and A. Bahai, “Energy-constrained modulation optimization,” *Wireless Communications, IEEE Transactions on*, vol. 4, no. 5, pp. 2349–2360, 2005. (Cit  en page 8.)
- [33] J. Karlsson and M. Skoglund, “Lattice-based source-channel coding in wireless sensor networks,” in *Communications (ICC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5. (Cit  en page 8.)
- [34] S.-T. Cheng, J.-S. Shih, C.-L. Chou, G.-J. Horng, and C.-H. Wang, “Hierarchical distributed source coding scheme and optimal transmission scheduling for wireless sensor networks,” *Wireless personal communications*, vol. 70, no. 2, pp. 847–868, 2013. (Cit  en page 8.)
- [35] X. Liu, “A survey on clustering routing protocols in wireless sensor networks,” *Sensors*, vol. 12, no. 8, pp. 11 113–11 153, 2012. [Online]. Available : <http://www.mdpi.com/1424-8220/12/8/11113> (Cit  en pages 8, 11, 12, 13, 15, 36 et 58.)
- [36] J. Yu, Y. Qi, G. Wang, and X. Gu, “A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution,” *International Journal of Electronics and Communications*, vol. 66, no. 1, pp. 54 – 61, 2012. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S1434841111001312> (Cit  en pages 8, 58 et 75.)
- [37] M. Zorzi and R. R. Rao, “Geographic random forwarding (geraf) for ad hoc and sensor networks : Multihop performance,” *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct. 2003. [Online]. Available : <http://dx.doi.org/10.1109/TMC.2003.1255648> (Cit  en pages 8 et 10.)

- [38] Ö. B. Akan and I. F. Akyildiz, “Event-to-sink reliable transport in wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 5, pp. 1003–1016, 2005. (Cité en page 9.)
- [39] C. Wang, K. Sohraby, B. Li, M. Daneshmand, and Y. Hu, “A survey of transport protocols for wireless sensor networks,” *Ieee Network*, vol. 20, no. 3, pp. 34–40, 2006. (Cité en page 9.)
- [40] L. Q. Tao and F. Q. Yu, “Ecoda : enhanced congestion detection and avoidance for multiple class of traffic in sensor networks,” *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 1387–1394, 2010. (Cité en page 9.)
- [41] P. Xie, Z. Zhou, Z. Peng, J.-H. Cui, and Z. Shi, “Sdrt : A reliable data transport protocol for underwater sensor networks,” *Ad Hoc Networks*, vol. 8, no. 7, pp. 708 – 722, 2010. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S157087051000017X> (Cité en page 9.)
- [42] J. Paek and R. Govindan, “Rcrt : Rate-controlled reliable transport protocol for wireless sensor networks,” *ACM Trans. Sen. Netw.*, vol. 7, no. 3, pp. 20 :1–20 :45, oct 2010. [Online]. Available : <http://doi.acm.org/10.1145/1807048.1807049> (Cité en page 9.)
- [43] N. Kimura and S. Latifi, “A survey on data compression in wireless sensor networks,” in *Information Technology : Coding and Computing, 2005. ITCC 2005. International Conference on*, vol. 2, April 2005, pp. 8–13 Vol. 2. (Cité en page 9.)
- [44] L. Xiang, J. Luo, and A. Vasilakos, “Compressed data aggregation for energy efficient wireless sensor networks,” in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*. IEEE, 2011, pp. 46–54. (Cité en page 9.)
- [45] F. Marcelloni and M. Vecchio, “Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization,” *Information Sciences*, vol. 180, no. 10, pp. 1924 – 1941, 2010, special Issue on Intelligent Distributed Information Systems. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S0020025510000496> (Cité en page 9.)
- [46] X. Xu, M. Li, X. Mao, S. Tang, and S. Wang, “A delay-efficient algorithm for data aggregation in multihop wireless sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 1, pp. 163–175, 2011. (Cité en page 9.)

- [47] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure data aggregation in wireless sensor networks," *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 3, pp. 1040–1052, 2012. (Cité en page 9.)
- [48] E. H. Callaway Jr, *Wireless sensor networks : architectures and protocols*. CRC press, 2004. (Cité en pages 9 et 11.)
- [49] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2/3, pp. 169–185, 2002. (Cité en page 10.)
- [50] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion : a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 56–67. (Cité en page 10.)
- [51] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*. ACM, 2002, pp. 22–31. (Cité en page 10.)
- [52] B. N. Karp and H. T. Kung, "Gpsr : Greedy perimeter stateless routing for wireless networks," in *the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, August 2000, pp. 243–254. (Cité en pages 10, 75 et 77.)
- [53] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing : a recursive data dissemination protocol for wireless sensor networks," UCLA Computer Science Department, Tech. Rep., 2001. (Cité en pages 10 et 75.)
- [54] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, vol. 1. IEEE, 2002, pp. 350–355. (Cité en page 10.)
- [55] P. Bighnaraj, D. Swades, and L. S. L. Jean-Daniel, "A greedy minimum energy consumption forwarding protocol for wireless sensor networks," in *Proceedings of the First international conference on COMMunication Systems And NETworks*, ser. COMSNETS'09. Piscataway, NJ, USA : IEEE Press, 2009, pp. 58–63. [Online]. Available : <http://dl.acm.org/citation.cfm?id=1702135.1702144> (Cité en page 10.)
- [56] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on wireless communication*, vol. 1, no. 4, pp. 660 – 670, October 2002. (Cité en pages 11, 12, 13, 14, 17, 18, 44, 46, 49, 58, 75 et 89.)

- [57] M. G. Loscrì V and M. S., “A two-levels hierarchy for low-energy adaptive clustering hierarchy,” in *62nd IEEE vehicular technology conference : VTC-2005-Fall*, September 2005, pp. 1809 – 1813. (Cité en pages 11, 14, 18 et 22.)
- [58] R. Randriatsiferana, R. Lorion, F. Alicalapa, and F. Harivelo, “Energy-efficient clustering algorithm based on energy variance for wireless sensor networks,” in *International Conference on Smart Communications in Network Technologies (SaCoNeT)*, vol. 01, 2013, pp. 1–5. (Cité en pages 12, 18, 58 et 79.)
- [59] X. Wu, G. Chen, and S. K. Das, “Avoiding energy holes in wireless sensor networks with nonuniform node distribution,” *IEEE Transactions on parallel and distributed systems*, vol. 19, no. 5, pp. 710–720, May 2008. (Cité en pages 12, 16 et 76.)
- [60] T. Arampatzis, J. Lygeros, S. Member, and S. Manesis, “A survey of applications of wireless sensors and wireless sensor networks,” in *Proc. 13 th Mediterranean Conference on Control and Automation, Limassol*, 2005, pp. 719–724. (Cité en pages 13 et 36.)
- [61] S. Mahajan and J. Malhotra, “Energy efficient control strategies in heterogeneous wireless sensor networks : A survey,” *International Journal of Computer Applications*, vol. 14, no. 6, pp. 31–37, February 2011, published by Foundation of Computer Science. (Cité en page 13.)
- [62] L. Xie, Y. Shi, Y. Hou, and A. Lou, “Wireless power transfer and applications to sensor networks,” *Wireless Communications, IEEE*, vol. 20, no. 4, pp. 140–145, August 2013. (Cité en page 13.)
- [63] K. Beydoun, “Conception d’un protocole de routage hierarchique pour les re-seaux de capteurs,” Ph.D. dissertation, PhD. Thesis, L’UFR des Sciences et Techniques de l’universite de Franche-Comte, Franche-Comte, 2009. (Cité en page 13.)
- [64] M. Chatterjee, S. K. Das, and D. Turgut, “Wca : A weighted clustering algorithm for mobile ad hoc networks,” *Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002. (Cité en pages 13, 14 et 15.)
- [65] C. Liu, K. Wu, and J. Pei, “A dynamic clustering and scheduling approach to energy saving in data collection from wireless sensor networks,” in *Sensor and Ad Hoc Communications and Networks, 2005. IEEE SECON 2005. 2005 Second Annual IEEE Communications Society Conference on*, Sept 2005, pp. 374–385. (Cité en pages 13, 14 et 15.)

- [66] K. Yanagihara, J. Taketsugu, K. Fukui, S. Fukunaga, S. Hara, and K.-i. Kitayama, "Eacle : Energy-aware clustering scheme with transmission power control for sensor networks," *Wireless Personal Communications*, vol. 40, no. 3, pp. 401–415, 2007. [Online]. Available : <http://dx.doi.org/10.1007/s11277-006-9199-2> (Cité en pages 13 et 14.)
- [67] L. Li, D. Shu-song, and W. Xiang-ming, "An energy efficient clustering routing algorithm for wireless sensor networks," *The journal of china universities of posts and telecommunications*, vol. 13, September 2006. (Cité en pages 13, 14 et 36.)
- [68] O. Younis and S. Fahmy, "Heed : a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004. (Cité en pages 13, 14, 19 et 75.)
- [69] S. Yi, J. Heo, Y. Cho, and J. Hong, "Peach : Power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2842–2852, 2007, network Coverage and Routing Schemes for Wireless Sensor Networks. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S0140366407002101> (Cité en pages 13, 14 et 15.)
- [70] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3, 2003, pp. 1713–1723. (Cité en page 14.)
- [71] Y. Jin, L. Wang, Y. Kim, and X. Yang, "Emc : An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks," *Computer Networks*, vol. 52, no. 3, pp. 542 – 562, 2008. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S1389128607002927> (Cité en page 14.)
- [72] S. Muruganathan, D. Ma, R. Bhasin, and A. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *Communications Magazine, IEEE*, vol. 43, no. 3, pp. S8–13, March 2005. (Cité en pages 14 et 28.)
- [73] J. Zhao and A. Erdogan, "A novel self-organizing hybrid network protocol for wireless sensor networks," in *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on*, June 2006, pp. 412–419. (Cité en page 14.)

- [74] S. Selvakenedy, S. Sinnappan, and Y. Shang, "A biologically-inspired clustering protocol for wireless sensor networks," *Computer Communications*, vol. 30, no. 14, pp. 2786–2801, 2007. (Cité en page 15.)
- [75] D. Karaboga, S. Okdem, and C. Ozturk, "Cluster based wireless sensor network routing using artificial bee colony algorithm," *Wirel. Netw.*, vol. 18, no. 7, pp. 847–860, Oct. 2012. [Online]. Available : <http://dx.doi.org/10.1007/s11276-012-0438-z> (Cité en page 15.)
- [76] B. Singh and D. Lobiyal, "A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks," *Human-centric Computing and Information Sciences*, vol. 2, no. 1, pp. 1–18, 2012. [Online]. Available : <http://dx.doi.org/10.1186/2192-1962-2-13> (Cité en page 15.)
- [77] L. Dehni, Y. Bennani, and F. Krief, "Lea2c : Low energy adaptive connectionist clustering for wireless sensor networks," in *MATA '05*, 2005, pp. 405–415. (Cité en pages 15 et 36.)
- [78] H. Taheri, P. Neamatollahi, O. M. Younis, S. Naghibzadeh, and M. H. Yaghmaee, "An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1469 – 1481, 2012. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S1570870512000753> (Cité en pages 15 et 20.)
- [79] T. Gao, R. Jin, J. Song, T. Xu, and L. Wang, "Energy-efficient cluster head selection scheme based on multiple criteria decision making for wireless sensor networks," *Wireless Personal Communications*, vol. 63, no. 4, pp. 871–894, 2012. [Online]. Available : <http://dx.doi.org/10.1007/s11277-010-0172-8> (Cité en page 15.)
- [80] L. Guo, Q. Li, and F. Chen, "A novel cluster-head selection algorithm based on hybrid genetic optimization for wireless sensor networks," *Journal of Networks*, vol. 6, no. 5, pp. 815–821, may 2011. (Cité en page 15.)
- [81] H. Zhang and A. Arora, "Gs3 : Scalable self-configuration and self-healing in wireless sensor networks," *Comput. Netw.*, vol. 43, no. 4, pp. 459–480, Nov. 2003. [Online]. Available : [http://dx.doi.org/10.1016/S1389-1286\(03\)00354-2](http://dx.doi.org/10.1016/S1389-1286(03)00354-2) (Cité en page 15.)
- [82] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2001, pp. 1028–1037 vol.2. (Cité en page 15.)

- [83] H. O. Tan and I. Körpeoğlu, “Power efficient data gathering and aggregation in wireless sensor networks,” *SIGMOD Rec.*, vol. 32, no. 4, pp. 66–71, Dec. 2003. [Online]. Available : <http://doi.acm.org/10.1145/959060.959072> (Cité en page 15.)
- [84] S. Lindsey and C. S. Raghavendra, “Pegasis : Power-efficient gathering in sensor information systems,” in *Aerospace conference proceedings, 2002. IEEE*, vol. 3. IEEE, 2002, pp. 3–1125. (Cité en pages 15, 16 et 30.)
- [85] A. Manjeshwar and D. P. Agrawal, “Teen : A routing protocol for enhanced efficiency in wireless sensor networks,” in *15th International Parallel and Distributed Processing Symposium*. IEEE, April 2000, pp. 2009 – 2015. (Cité en page 15.)
- [86] —, “Apteen : A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks,” in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, ser. IPDPS '02. Washington, DC, USA : IEEE Computer Society, 2002, pp. 48–56. [Online]. Available : <http://dl.acm.org/citation.cfm?id=645610.662036> (Cité en pages 15, 16 et 75.)
- [87] P. Ding, J. Holliday, and A. Celik, “Distributed energy-efficient hierarchical clustering for wireless sensor networks,” in *Distributed Computing in Sensor Systems*. Springer, 2005, pp. 322–339. (Cité en pages 15 et 20.)
- [88] C. Zhu, C. Zheng, L. Shu, and G. Han, “A survey on coverage and connectivity issues in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619 – 632, 2012, simulation and Testbeds. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S1084804511002323> (Cité en page 15.)
- [89] N. M. Freris, H. Kowshik, and P. Kumar, “Fundamentals of large sensor networks : Connectivity, capacity, clocks, and computation,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1828–1846, 2010. (Cité en page 15.)
- [90] J. Li and P. Mohapatra, “Analytical modeling and mitigation techniques for the energy hole problem in sensor networks,” *Pervasive Mob. Comput.*, vol. 3, no. 3, pp. 233–254, Jun. 2007. [Online]. Available : <http://dx.doi.org/10.1016/j.pmcj.2006.11.001> (Cité en page 15.)
- [91] R. Randriatsiferana, R. Lorion, F. Alicalapa, and F. Harivelo, “Forwarding and routing stateless multi-hop protocol for wireless sensor networks,” in *ICN 2012 : The Eleventh International Conference on Networks*, ser. IARIA, 2012, February 2012, pp. 165–168. (Cité en pages 15 et 76.)

- [92] A.-F. Liu, P.-H. Zhang, and Z.-G. Chen, "Theoretical analysis of the lifetime and energy hole in cluster based wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 71, no. 10, pp. 1327 – 1355, 2011. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S0743731511000992> (Cité en page 16.)
- [93] M. Ye, C. Li, G. Chen, and J. Wu, "Eecs : An energy efficient clustering scheme in wireless sensor networks," in *24th IEEE International Performance, Computing, and Communications Conference.*, ser. IPCCC '2005., April 2005, pp. 535 – 540. (Cité en pages 16 et 25.)
- [94] G. Chen, C. Li, M. Ye, and J. Wu, "An unequal cluster-based routing protocol in wireless sensor networks," *Wireless Netw.*, vol. 15, no. 2, pp. 193–207, Feb. 2009. [Online]. Available : <http://dx.doi.org/10.1007/s11276-007-0035-8> (Cité en pages 16, 27 et 84.)
- [95] M. Ali, T. Dey, and R. Biswas, "Aleach : Advanced leach routing protocol for wireless microsensor networks," in *Electrical and Computer Engineering, 2008. ICECE 2008. International Conference on*, Dec 2008, pp. 909–914. (Cité en pages 18 et 36.)
- [96] M. Farooq, A. Dogar, and G. Shah, "Mr-leach : Multi-hop routing with low energy adaptive clustering hierarchy," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, July 2010, pp. 262–268. (Cité en page 18.)
- [97] G. Rana, H. Zhang, and S. Gong, "Improving on leach protocol of wireless sensor networks using fuzzy logic," *Journal of Information & Computational Science*, vol. 7, pp. 67–775, 2010. (Cité en page 18.)
- [98] N. Wang and H. Zhu, "An energy efficient algorithm based on leach protocol," in *International Conference on Computer Science and Electronics Engineering (ICCSEE)*, vol. 2. IEEE Comp. Soc, March 2012, pp. 339 – 342. (Cité en page 18.)
- [99] S. Soro and W. B. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering," in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*. IEEE, 2005, pp. 8–pp. (Cité en page 24.)
- [100] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, Nov 2005, pp. 8 pp.–604. (Cité en pages 27 et 75.)

- [101] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless sensor networks : A survey on the state of the art and the 802.15.4 and zigbee standards," *Computer Communication*, no. 30, pp. 1655–1695, 2007. (Cit  en page 36.)
- [102] C. O. Math na, T. O'Donnell, R. V. Martinez-Catala, J. Rohan, and B. O'Flynn, "Energy scavenging for long-term deployable wireless sensor networks," *Talanta*, vol. 75, no. 3, pp. 613 – 623, 2008, special Section : Remote Sensing. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S0039914007008612> (Cit  en page 36.)
- [103] M. G. Rashed, M. H. Kabir, and S. E. Ullah, "An energy efficient protocol for cluster based heterogeneous wireless sensor network," *International Journal of Distributed and Parallel Systems*, vol. 2, no. 2, pp. 54–60, 2011. (Cit  en page 36.)
- [104] L. Ramachandran, M. Kapoor, A. Sarkar, and A. Aggarwal, "Clustering algorithms for wireless ad hoc networks," in *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ser. DIALM '00. New York, NY, USA : ACM, 2000, pp. 54–63. [Online]. Available : <http://doi.acm.org/10.1145/345848.345860> (Cit  en page 36.)
- [105] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, sep 2007. [Online]. Available : <http://doi.acm.org/10.1145/1274858.1274870> (Cit  en page 36.)
- [106] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *the 33rd Hawaii International Conference on System Sciences (HICSS)*, January 2000, pp. 1–10. (Cit  en page 37.)
- [107] Nurhayati, S. H. Choi, and K. O. Lee, "A cluster based energy efficient location routing protocol in wireless sensor networks," *International journal of computers and communications*, vol. 5, pp. 67–74, 2011. (Cit  en page 37.)
- [108] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 40–50, Mar 2002. (Cit  en pages 39 et 120.)
- [109] R. Kacimi, "Techniques de conservation d' nergie pour les r seaux de capteurs sans fil," Ph.D. dissertation, 2009. (Cit  en page 39.)

- [110] W. A. Hendricks and K. W. Robey, "Sampling distribution of the coefficient of variation," *The Annals of Mathematical Statistics*, vol. 7, no. 3, pp. 129–132, 09 1936. [Online]. Available : <http://dx.doi.org/10.1214/aoms/1177732503> (Cité en page 40.)
- [111] B. Iglewicz and R. H. Myers, "Comparisons of approximations to the percentage points of the sample coefficient of variation," *Technometrics*, vol. 12, no. 1, pp. 166–169, 1970. [Online]. Available : <http://www.jstor.org/stable/1267363> (Cité en page 40.)
- [112] M. G. Vangel, "Confidence intervals for a normal coefficient of variation," *The American Statistician*, vol. 50, no. 1, pp. 21–26, 1996. [Online]. Available : <http://www.jstor.org/stable/2685039> (Cité en page 40.)
- [113] L. H. Koopmans, D. B. Owen, and J. I. Rosenblatt, "Confidence intervals for the coefficient of variation for the normal and log normal distributions," *Biometrika*, vol. 51, pp. 25–32, 1964. (Cité en page 41.)
- [114] R. Mahmoudvand, H. Hassani, and R. Wilson, "Is the sample coefficient of variation a good estimator for the population coefficient of variation?" University Library of Munich, Germany, MPRA Paper 6106, Sep. 2007. [Online]. Available : <http://ideas.repec.org/p/pra/mprapa/6106.html> (Cité en page 41.)
- [115] J. Forkman, "Estimator and tests for common coefficients of variation in normal distributions." *Communications in Statistics - Theory and Methods*, vol. 38, pp. 233–251, 2009. (Cité en page 41.)
- [116] J. Hoseini and A. Mohammadi, "Estimator and tests for coefficient of variation in uniform distribution," *J Biomet Biostat*, vol. 3, p. 149, 2012. (Cité en page 41.)
- [117] W. Panichkitkosolkul, "Confidence intervals for the coefficient of variation in a normal distribution with a known population mean," *Journal of Probability and Statistics*, vol. 2013, p. 11, 2013. (Cité en page 41.)
- [118] T.-J. Chan, C.-M. Chen, Y.-F. Huang, J.-Y. Lin, and T.-R. Chen, "Optimal cluster number selection in ad-hoc wireless sensor networks," *WTOC*, vol. 7, no. 8, pp. 837–846, Aug. 2008. (Cité en page 44.)
- [119] R. Randriatsiferana, R. Lorion, F. Alicalapa, and A.-M. Mohammed., "A clustering algorithm based on energy variance and coverage density in centralized hierarchical wireless sensor networks," in *AFRICON 2013, International Conference on*, vol. 01, 2013, pp. 1–5. (Cité en page 46.)

- [120] R. Kershner, “The number of circles covering a set,” *American Journal of Mathematics*, vol. 61, no. 3, pp. 665–671, 1939. [Online]. Available : <http://www.jstor.org/stable/2371320> (Cité en page 47.)
- [121] S. Verblunsky, “On the least number of unit circles which can cover a square,” *J. London Math. Soc.*, vol. 24, pp. 164–170, 1949. [Online]. Available : <http://jlm.oxfordjournals.org/content/s1-24/3/164> (Cité en page 47.)
- [122] G. Tóth, “Covering the plane by convex discs,” *Acta Mathematica Academiae Scientiarum Hungarica*, vol. 23, no. 1-2, pp. 263–270, 1972. [Online]. Available : <http://dx.doi.org/10.1007/BF01889922> (Cité en page 47.)
- [123] G. Tóth and A. Florian, “Covering of the plane by discs,” *Geometriae Dedicata*, vol. 16, no. 3, pp. 315–333, 1984. [Online]. Available : <http://dx.doi.org/10.1007/BF00147874> (Cité en page 47.)
- [124] G. F. Tóth, “Best partial covering of a convex domain by congruent circles of a given total area,” *Discrete Comput. Geom.*, vol. 38, no. 2, pp. 259–271, Sep. 2007. (Cité en page 48.)
- [125] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, ser. SenSys ’03. New York, NY, USA : ACM, 2003, pp. 28–39. [Online]. Available : <http://doi.acm.org/10.1145/958491.958496> (Cité en page 48.)
- [126] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, “Peas : A robust energy conserving protocol for long-lived sensor networks,” in *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on.* IEEE, 2003, pp. 28–37. (Cité en page 48.)
- [127] P. K. Agarwal and C. M. Procopiuc, “Exact and approximation algorithms for clustering,” in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’98. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 1998, pp. 658–667. [Online]. Available : <http://dl.acm.org/citation.cfm?id=314613.315040> (Cité en page 49.)
- [128] —, “Exact and approximation algorithms for clustering,” *Algorithmica*, vol. 33, no. 2, pp. 201–226, 2002. [Online]. Available : <http://dx.doi.org/10.1007/s00453-001-0110-y> (Cité en page 49.)
- [129] T. Acharya and G. Paul, “Maximum lifetime broadcast communications in cooperative multihop wireless ad hoc networks : Centralized and

- distributed approaches,” *Ad Hoc Networks*, vol. 11, no. 6, pp. 1667 – 1682, 2013. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S1570870513000322> (Cité en page 67.)
- [130] B. N. Karp, “Geographic routing for wireless networks,” Ph.D. Dissertation, Harvard University, Cambridge, MA, October 2000. (Cité en pages 75 et 77.)
- [131] Y. Bolian, S. Hongchi, and S. Yi, “A two-level topology control strategy for energy efficiency in wireless sensor networks,” in *Parallel and Distributed Systems, 11th International Conference on*, vol. 2, Washington, DC, USA. IEEE Computer, November 2005, pp. 358–362. (Cité en pages 75 et 76.)
- [132] V. Mhatre and C. Rosenberg, “Design guidelines for wireless sensor networks : communication, clustering and aggregation,” *Ad Hoc Networks*, vol. 2, no. 1, pp. 45 – 63, 2004. [Online]. Available : <http://www.sciencedirect.com/science/article/pii/S1570870503000477> (Cité en page 76.)
- [133] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, “Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks,” in *Proceedings of the Workshop on Real-world Wireless Sensor Networks*, ser. REALWSN '08. New York, NY, USA : ACM, 2008, pp. 1–5. [Online]. Available : <http://doi.acm.org/10.1145/1435473.1435475> (Cité en page 78.)
- [134] W. Wang, B. Wang, Z. Liu, L. Guo, and W. Xiong, “A cluster-based and tree-based power efficient data collection and aggregation protocol for wireless sensor networks,” *Information technology journal*, vol. 10, no. 3, pp. 557–564, 2011. (Cité en pages 80 et 84.)
- [135] M. Liu, J. Cao, G. Chen, and X. Wang, “An energy-aware routing protocol in wireless sensor networks,” *Sensors*, vol. 9, no. 1, pp. 445–462, 2009. (Cité en page 84.)
- [136] D. Gay, P. Levis, and D. Culler, “Software design patterns for tinys,” *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, Sep. 2007. [Online]. Available : <http://doi.acm.org/10.1145/1274858.1274860> (Cité en page 90.)
- [137] “Ettx 2009 : The first european tinys technology exchange,” Fev 2009. [Online]. Available : <https://sites.google.com/site/ettx2009/welcome> (Cité en page 90.)
- [138] D. Gay, M. Welsh, P. Levis, E. Brewer, R. V. Behren, and D. Culler, “The nesc language : A holistic approach to networked embedded systems,” in *In Proceedings of Programming Language Design and Implementation (PLDI)*, 2003, pp. 1–11. (Cité en page 90.)

- [139] *Differences between Tmote sky and Telos Revision B.* [Online]. Available : http://automatica.dei.unipd.it/public/Schenato/PSC/2010_2011/gruppo4-Building_termo_identification/Bibliografia20Casuale/moteiv-an-001.pdf (Cit  en pages 91, 95 et 111.)
- [140] *Datasheet tmote-sky : Ultra low power IEEE 802.15.4 compliant wireless sensor module*, 2006. [Online]. Available : <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf> (Cit  en page 91.)
- [141] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, and D. Culler., “Flexible hardware abstraction for wireless sensor networks,” in *Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on*, Jan 2005, pp. 145–157. (Cit  en page 93.)
- [142] V. Handziski, J. Polastre, J.-H. Hauer, C. Sharp, A. Wolisz, D. Culler, and D. Gay, “Hardware abstraction architecture,” TEP : TinyOS Enhancement Proposal 2. (Cit  en page 93.)
- [143] *MSP430x1xx Family : User’s Guide*, 2006. [Online]. Available : <http://www.ti.com/lit/ug/slau049f/slau049f.pdf> (Cit  en page 94.)
- [144] J. H. Davies, *MSP430 Microcontroller Basics*. [Online]. Available : <http://www.alegus.ru/Literatura/MSP430%20Microcontroller%20Basics.pdf> (Cit  en pages 94, 95 et 103.)
- [145] Q. Wang, Y. Zhu, and L. Cheng, “Reprogramming wireless sensor networks : challenges and approaches,” *Network, IEEE*, vol. 20, no. 3, pp. 48–55, May 2006. (Cit  en page 96.)
- [146] C. Kitchin and L. Counts, *A designer’s guide to instrumentation amplifiers*, ser. 3rd Edition, 2006. (Cit  en pages 98 et 101.)
- [147] *Data Sheet AD627 :Micropower, Single- and Dual-Supply, Rail-to-Rail Instrumentation Amplifier*. [Online]. Available : http://www.analog.com/static/imported-files/data_sheets/AD627.pdf (Cit  en pages 98, 99 et 100.)

Optimisation énergétique des protocoles de communication du RCSF

Résumé : Pour augmenter la durée de vie des réseaux de capteurs sans fil, une solution est d'améliorer l'efficacité énergétique des protocoles de communication. Le regroupement des noeuds du réseau de capteurs sans fil en cluster est l'une des meilleures méthodes. Cette thèse présente propose plusieurs améliorations en modifiant les paramètres du protocole de référence LEACH. Pour améliorer la distribution énergétique des "cluster-heads", nous proposons deux protocoles de clustering centralisés k-LEACH et sa version optimisée k-LEACH-VAR. Un algorithme distribué, appelé e-LEACH, est également proposé pour réduire l'échange d'information périodique entre les noeuds et la station de base lors de l'élection des "cluster-heads". Par ailleurs, le concept l'équilibrage énergétique est introduit dans les métriques d'élection pour éviter les surcharges des noeuds. Ensuite, nous présentons une version décentralisée de k-LEACH qui, en plus des objectifs précédents, intègre la consommation d'énergie globale du réseau. Ce protocole, appelé, k-LEACH-C2D, vise également à favoriser la scalabilité du réseau. Pour renforcer ce dernier et l'autonomie des réseaux, les deux protocoles de routage "multi-hop" probabiliste, dénotés FRSM et CB-RSM construisent des chemins élémentaires entre les "cluster-heads" et la station de base. Le protocole CB-RSM forme une hiérarchie des "cluster-heads" pendant la phase de formation des clusters, en mettant un accent sur l'auto-ordonnancement et l'auto-organisation entre les "cluster-heads" pour rendre les réseaux le plus "scalable". Ces différents protocoles reposent sur l'idée de base que les noeuds ayant l'énergie résiduelle la plus élevée et la plus faible variance de consommation de l'énergie deviennent "cluster-head". Nous constatons le rôle central de la consommation du noeud dans nos différentes propositions. Ce point fera l'objet de la dernière partie de cette thèse. Nous proposons une méthodologie pour caractériser expérimentalement la consommation d'un noeud. Les objectifs visent à mieux appréhender la consommation pour différentes séquences d'état du noeud. Enfin, nous proposons un modèle global de la consommation du noeud.

Mots clés : Efficacité énergétique, durée de vie, sélection des cluster-heads, variance d'énergie, équilibrage énergétique, auto-ordonnancement, scalabilité.

Energy optimization of communication protocols of the WSN

Abstract : To increase the lifetime of wireless sensor networks, a solution is to improve the energy efficiency of the communication's protocol. The grouping of nodes in the wireless sensor network clustering is one of the best methods. This thesis proposes several improvements by changing the settings of the reference protocol LEACH. To improve the energy distribution of "cluster-heads", we propose two centralized clustering protocols LEACH and k-optimized version k-LEACH-VAR. A distributed algorithm, called e-LEACH, is proposed to reduce the periodic exchange of information between the nodes and the base station during the election of "cluster-heads". Moreover, the concept of energy balance is introduced in metric election to avoid overloading nodes. Then we presented a decentralized version of k-LEACH, which in addition to the previous objectives, integrates the overall energy consumption of the network. This protocol, called k-LEACH-C2D, also aims to promote the scalability of the network. To reinforce the autonomy and networks, both routing protocols "multi-hop" probability, denoted CB-FRSM and RSM build elementary paths between the "cluster-heads" and elected the base station. The protocol, CB-RSM, forms a hierarchy of "cluster-heads" during the training phase clusters, with an emphasis on self-scheduling and self-organization between "cluster-heads" to make the networks more scalable. These protocols are based on the basic idea that the nodes have the highest residual energy and lower variance of energy consumption become "cluster-head". We see the central role of consumption of the node in our proposals. This point will be the last part of this thesis. We propose a methodology to characterize experimentally the consumption of a node. The objectives are to better understand the consumption for different sequences of the node status. In the end, we propose a global model of the consumption of the node.

Keywords : energy efficiency, lifetime, cluster-head selection, variance energy, energy distribution, energy balancing, auto-scheduling, scalability.
